

宁夏回族自治区教育厅中小学教材审查委员会审定

# 信息技术

## LOGO 语言与机器人

“LOGO 语言与机器人整合”实验教材

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

#### 图书在版编目(CIP)数据

信息技术. LOGO 语言与机器人 / 杨宏轩, 王英编著. —北京: 电子工业出版社, 2015.8  
ISBN 978-7-121-26141-1

I. ①信… II. ①杨… ②王… III. ①计算机课—小学—教材 IV. ①G624.581

中国版本图书馆 CIP 数据核字 (2015) 第 111473 号

策划编辑：董晓梅

责任编辑：马 杰

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱

邮编：100036

开 本：787×1092 1/16 印张：9.25

字数：199 千字

版 次：2015 年 8 月第 1 版

印 次：2015 年 8 月第 1 次印刷

定 价：16.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，  
联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：(010) 88258888。



# 编 者 的 话

程序设计和机器人都是中小学信息技术学科重要的教学内容。程序是智能机器人的灵魂，机器人需要用计算机语言编程控制。以机器人的搭建、调试、运行作为载体，让学生学习算法、物理、机械等科学知识，初步理解程序设计的思想和方法，培养逻辑思维能力和动手能力，是全面提升中小学生综合素质的有效途径。

LOGO 语言是最适合小学生学习的程序设计语言之一。用 LOGO 语言编写控制机器人的程序，在仿真环境下进行程序的调试和修改，然后下载到学生动手搭建的实物机器人上，控制机器人去完成任务。在这个过程中，利用学习机器人作为程序设计 (LOGO 语言) 学习的载体，通过机器人的行为来检验程序设计的结果，将从根本上解决程序设计在小学信息技术教学中存在的抽象、单调、学习兴趣低等问题。因此，LOGO 语言、虚拟仿真机器人及实物机器人整合，是小学计算机程序设计教学的最佳模式。

目前，小学的程序设计与机器人教学多为独立进行，程序设计与机器人整合教学的实验尚处于探索阶段，进行“虚实结合”的程序设计与机器人整合教学实验，将是中小学信息技术课程教学内容和教学方式的创新探索。

这本实验教材采用与宁夏义务教育信息技术教材相同的体例，有利于实验区教师、学生快速进入教学、学习状态，结合杭州萝卜圈网络科技有限公司为实验配套开发的 IRobotQ3D 平台和机器人(套件)，适合不同起点的学校和学生使用。

编写这本教材，尽管我们已经非常努力，但不足和缺憾在所难免，恳切希望广大师生提出宝贵意见，以便进一步修改与完善。

课题组

2015 年 8 月

## “LOGO 语言与机器人整合” 课题组

组 长： 岳维鹏

副组长： 黄建忠 杨宏轩

成 员： 张建勇 赵志坚 王 英 赵志新  
郭丽红 徐永惠 杨梅林 秦春娟

编 著： 杨宏轩 王 英

协作单位： 杭州萝卜圈网络技术有限公司



# LOGO 语言与机器人

## 第 1 单元 初识 LOGO /2

- 一、启动 LOGO /2
- 二、LOGO 的窗口 /3
- 三、退出 LOGO /5
- 四、LOGO 语言的基本命令 /6
- 巩固练习 /14
- 实践活动 /16
- 阅读材料 LOGO 语言简介 /17

## 第 2 单元 LOGO 的过程 /18

- 一、过程的定义和保存 /18
- 二、使用重复命令 /23
- 三、带参数的过程 /25
- 四、调用过程 /29
- 巩固练习 /32
- 实践活动 /34
- 阅读材料 用计算机程序解决问题 /35

## 第 3 单元 与机器人见面 /36

- 一、初步认识机器人 /36
- 二、机器人是怎样工作的 /38
- 三、机器人仿真平台 /41
- 四、学习实践 /49
- 巩固练习 /53
- 阅读材料 机器人的发展 /54

## 第 4 单元 搭建机器人 /56

- 一、构建机器人部件 /57
- 二、搭建仿真机器人 /64
- 三、学习实践 /68
- 巩固练习 /69

## 第 5 单元 安全出站 /71

- 一、熟悉 IRobotQ3D 的编程环境 /72
- 二、体验 IRobotQ3D 仿真环境编程 /74





三、学习实践 /77

巩固练习 /80

## 第6单元 新手上路 /82

一、超声波传感器 /82

二、在 IRobotQ3D 中给机器人  
添加超声波传感器 /82

三、仿真体验使用超声波传感器 /89

四、学习实践 /93

巩固练习 /94

阅读材料 机器人常用的几种传感器  
/95

## 第7单元 小试牛刀 /96

一、灰度传感器 /96

二、在 IRobotQ3D 中给机器人添  
加灰度传感器 /96

三、仿真体验使用灰度传感器 /103

四、学习实践 /107

巩固练习 /108

## 第8单元 展示技艺 /110

一、障碍传感器和触碰传感器 /110

二、在 IRobotQ3D 中给机器人添加障碍

传感器和触碰传感器 /111

三、仿真体验使用障碍传感器和触碰  
传感器 /114

四、学习实践 —— “学习者号” 走迷  
宫 /119

巩固练习 /120

## 第9单元 一鼓作气 /121

一、仿真体验模块化程序设计 /121

二、学习实践 /129

巩固练习 /130

## 第10单元 从LOGO语言编程到可视化语 言编程 /132

一、可视化程序设计 /132

二、熟悉 IRobotQ3D 的 VPL 编程环境  
/132

三、体验 VPL 编程 /135

四、学习实践 /140

巩固练习 /140

综合实践活动 /141

学习评价 /142



# LOGO 语言与机器人

LOGO 语言是一种适合小学生学习计算机程序设计的入门语言，学会 LOGO 语言后，就可以利用它来指挥计算机为我们做事。机器人集合了计算机、声、光、电等多种技术，涉及传感器、计算机软硬件、人工智能、自动控制等高新科技知识，是现代信息技术和先进制造技术相结合的产物，它代表着一个国家高新技术发展的综合水平。程序设计和机器人都是小学信息技术学习的内容。

智能机器人是在微型计算机(单片机)的程序指挥下工作的。因此，机器人与程序设计密切相关。本册教材将机器人知识和 LOGO 语言整合在一起，把机器人作为学习程序设计的载体，程序设计学习是否达到预期的目标，可以通过机器人(仿真和真实)的行为来检验，使抽象、单调的程序设计变得直观而具体。让我们在搭建机器人、为机器人编写程序、调试、运行机器人的过程中，学到程序设计和机器人的基础知识，学会科学、高效的思维方式，提高动手实践和创新能力。

本册教材共 10 个单元。第 1~4 单元作为基础铺垫，其中 1、2 单元学习 LOGO 语言的基础知识，3、4 单元学习机器人基础知识并学会搭建机器人。从第 5 单元到第 9 单元，我们将以一名“驾驶员”的身份，在不断完善我们搭建的“学习者号”机器人的基础上，“驾驶”它完成“安全出站”“精准停靠”“遵守交规”“魔幻寻宝”系列任务，经历将一台机器逐渐变成智能机器人的过程。在第 10 单元，将总结我们积累的编程经验并过渡到可视化程序设计，为今后学习打下基础。

通过本模块的学习，你将：

- ❖ 知道计算机程序设计的作用，领会程序设计的思想和方法。
- ❖ 掌握 LOGO 语言的常用命令，能使用 LOGO 语言编写简单的计算机程序，体验用计算机程序指挥计算机做事的乐趣。
- ❖ 理解基本的逻辑思维、物理、机械和算法等学科知识，实现计算机程序设计与机器人学习的整合。
- ❖ 了解机器人的应用，学会搭建教学机器人(仿真、真实)。
- ❖ 会使用 LOGO 命令指挥教学机器人(仿真、真实)完成预定的动作。
- ❖ 了解传感器的作用，会为教学机器人(仿真、真实)配置传感器获取信息。





# 第 1 单元 初识 LOGO

看到图 1.1 所示两幅漂亮的图案，你是不是想问，它们也是用画图软件画出来的吗？

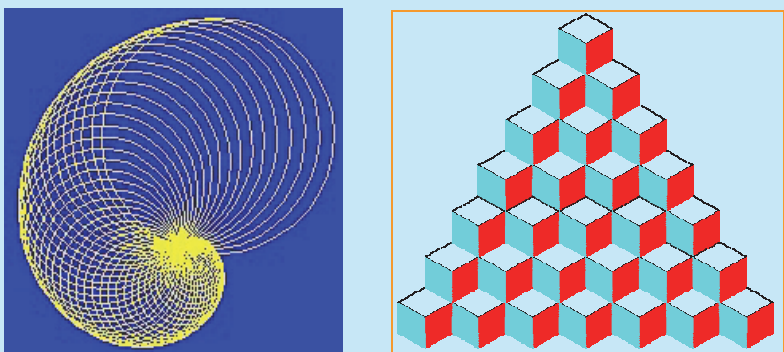






图 1.1 计算机绘制的图案

告诉你吧，这两幅图案不是用“画图”软件画出来的，而是计算机执行 LOGO 语言编写的程序自动绘制出来的。

程序就是用计算机语言编写的，指示计算机每一步动作的一组指令。用于编写程序的计算机语言称为编程语言。LOGO 语言是一种与自然语言非常接近的编程语言，它通过指挥小海龟“绘图”的方式来学习编程，非常适合小学生学习程序设计。LOGO 语言的版本很多，功能和使用方法基本相同，我们学习使用的是 Windows 系统下的 PC LOGO 4.0 中文版，以下简称 LOGO。

## 一、启动 LOGO

启动 LOGO 的方法与启动一般 Windows 应用程序的方法类似。单击  按钮，打开“开始”菜单，单击  所有程序 命令打开程序列表，单击  PC Logo 4.0 双语版 打开该文件夹下的程序列表，最后单击  PC Logo 4.0 中文版，如图 1.2 所示，即可启动 LOGO。

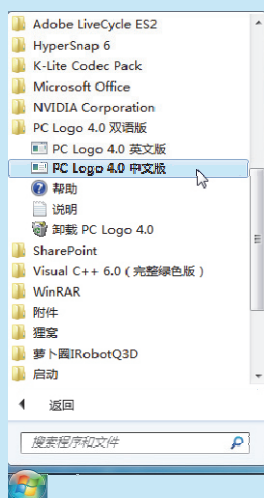


图 1.2 启动 LOGO



## 试试看

用你习惯的方法启动 LOGO，并和同学交流启动 LOGO 的常用方法。



## 二、LOGO 的窗口

启动 LOGO 后，桌面显示 LOGO 窗口，如图 1.3 所示。

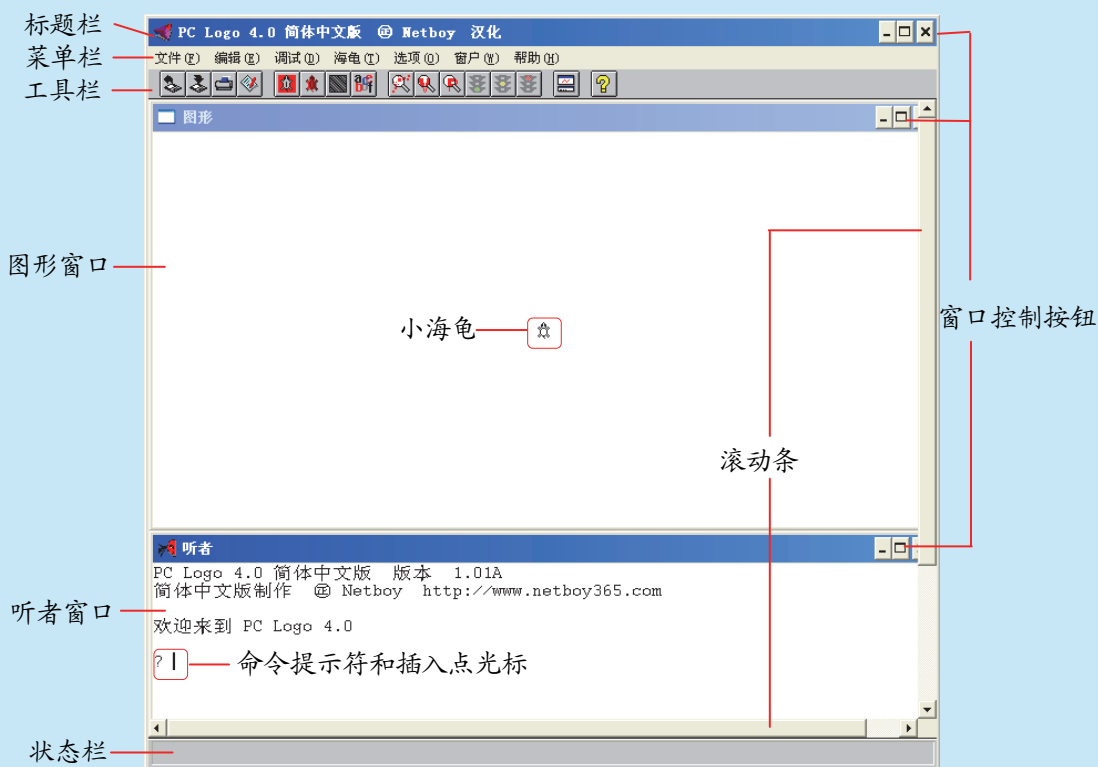


图 1.3 LOGO 窗口

与我们熟悉的其他 Windows 应用程序窗口相仿，LOGO 窗口由“标题栏”“菜单栏”“工具栏”“状态栏”“窗口控制按钮”“滚动条”等组成，工具栏和状态栏中间是工作区。

图形窗口中有一个栩栩如生的小海龟：尖尖的脑袋、四只小脚和一条小尾巴。每次启动 LOGO 后，小海龟总是头向上出现在图形窗口的正中央，这个位置叫做“母位”，它是小海龟运动的起点，又被称为小海龟的“家”。小海龟的尾巴具有特异功能——那是一支神奇的“画笔”！小海龟在图形窗口中运动时，这支“画笔”能记录下它的运动轨迹，这些轨迹可以组成生动有趣、丰富多彩的图形，所以人们送给小海龟一个“绘画高手”的称号。图形窗口又称为绘图区，显然，绘图区就是小海龟



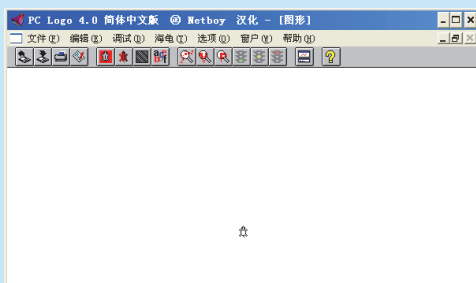


画画的“画布”了。图 1.1 就是小海龟用神奇“画笔”画出来的图画。

在 LOGO 图形窗口正中央待命的小海龟能“听懂” LOGO 语言，当我们在听者窗口提示符“？”的后面输入用 LOGO 语言编写的命令，小海龟就会在绘图区中做出相应的动作。

LOGO 有三种窗口方式，“图文混合显示方式”“全屏幕图形显示方式”和“全屏幕文字显示方式”。系统默认打开“图文混合显示方式”窗口(参见图 1.3)，它的工作区由图形窗口和听者窗口组成。听者窗口标题栏下方的区域是编写程序、向计算机发布命令的区域，在显示 LOGO 版本、网址和欢迎信息的后面，有一个开头是命令提示符“？”、其后闪动着插入点光标“|”的行，称为命令行。在提示符的后面输入指令，计算机(即听者)就根据命令指挥小海龟在图形窗口中动作。

全屏幕图形显示方式(全屏幕图形窗口方式)和全屏幕文字显示方式(全屏幕听者窗口方式)是为了便于操作，根据工作内容对默认窗口的简化界面，如图 1.4 所示。



全屏幕图形显示方式



全屏幕文字显示方式

图 1.4 LOGO 的另外两种窗口显示方式

LOGO 的三种窗口可以利用键盘快速进行切换。三种窗口对应的快捷键如表 1.1 所示。

表 1.1 三种窗口对应的快捷键

图文混合显示方式 (图形+听者窗口方式)	全屏幕文字显示方式 (全屏幕听者窗口方式)	全屏幕图形显示方式 (全屏幕图形窗口方式)
Shift + F2	Shift + F3	Shift + F4


PC LOGO 4.0 中文版是基于 Windows 操作系统的，我们可以像操作其他 Windows 应用程序窗口那样，改变 LOGO 窗口及其中两个小窗口的大小、位置和背景颜色。

### 说说看

LOGO 窗口与“画图”程序窗口有哪些相同之处？  
有哪些不同之处？



### 试试看

1. 在 LOGO 窗口中，用鼠标依次指向工具栏中的各个图标，查看系统提示，记下每个工具图标的名称和作用。（提示：可以利用菜单栏中的“帮助”命令。）
2. 单击 LOGO 窗口的最大化按钮，最大化 LOGO 窗口，将图形窗口和听者窗口完整地显示出来。然后试着改变一下图形窗口和听者窗口的大小。图 1.5 供参考。

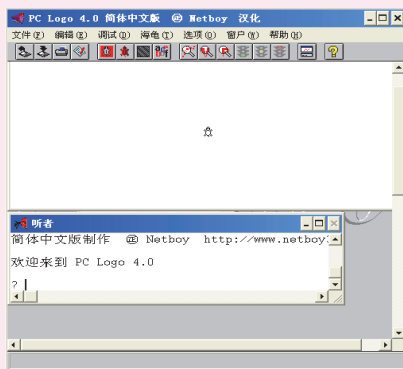


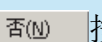


图 1.5 改变 LOGO 窗口中两个小窗口的大小



## 三、退出 LOGO

退出 LOGO 的方法和退出“画图”“写字板”等应用程序的方法相似。单击标题栏右端的关闭按钮或执行“文件”菜单中的“退出”命令，会弹出如图 1.6 所示的对话框，提醒我们确认关闭窗口操作：单击按钮，关闭窗口退出 LOGO；单击按钮，撤销关闭窗口的操作。

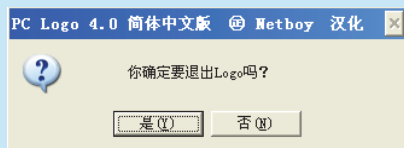


图 1.6 退出 LOGO 确认对话框

LOGO 还提供了一种独特的退出方式：在听者窗口命令提示符后输入“EXIT”（如图 1.7 所示）后按回车键，也会弹出图 1.6 所示对话框，要我们确认退出操作。



图 1.7 输入“EXIT”命令退出



**试试看**

分别用上述三种方法退出 LOGO。



## 四、LOGO 语言的基本命令

LOGO 语言的基本命令包括与移动有关的命令和与控制有关的命令。

### 1. LOGO 语言的命令格式

在 LOGO 语言中，“指挥”小海龟动作通常使用命令。LOGO 命令的一般格式是：  
命令 + 空格 + [参数]

其中：“命令”指命令名称，指挥小海龟怎样动作，通常用命令的英文单词缩写表示。如让小海龟前进的英文单词是 Forward，“前进”命令就用“FD”表示；“参数”说明命令的执行方式，如“前进”(FD)的“参数”是小海龟前进的距离，一般用“步长”表示。如果命令格式中的“参数”用中括号括起来，表示它是可选的，即有些命令可能没有参数。

### 2. LOGO 语言的移动命令

在图形窗口中，小海龟的基本动作有：前进(Forward)、后退(Back)、左转(Left)、右转(Right)等，它们对应的 LOGO 命令如表 1.2 所示。

表 1.2 LOGO 中的基本命令 1

命 令	格 式	作 用
前进(Forward)	FD 步长(前进的距离)	小海龟按当前的方向前进指定的距离，并画出线段
后退(Back)	BK 步长(后退的距离)	小海龟按当前的方向后退指定的距离，并画出线段
左转(Left)	LT 度数(转过的角度)	小海龟向左旋转指定的度数
右转(Right)	RT 度数(转过的角度)	小海龟向右旋转指定的度数

在 LOGO 的听者窗口中，输入命令后必须按回车键，命令才会被执行。

**做一做**

用 LOGO 命令指挥小海龟画出如图 1.8 所示的折线。

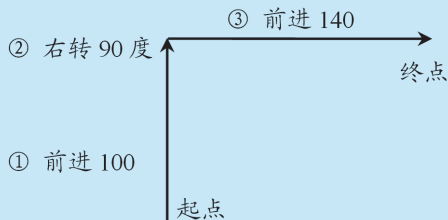


图 1.8 画折线

第 1 步：启动 LOGO。

第 2 步：在听者窗口的命令提示符“？”后输入命令“FD 100”，按回车键，小海龟在图形窗口中前进一段距离并画出一条线段，如图 1.9 所示。

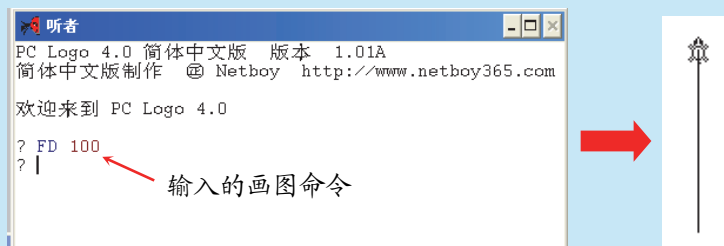


图 1.9 输入并执行“FD”命令及画图效果

### 小博士

输入的 LOGO 命令既可以是大大写字母,也可以是小写字母。例如,“FD 100”命令也可以写成“fd 100”,两种输入方式的执行效果相同。

在 LOGO 窗口中,输入命令后就被立即执行,这种运行方式称为“立即方式”或“命令方式”。



第 3 步：在新一行命令提示符“？”后输入“RT 90”，按回车键，小海龟在图形窗口中向右旋转 90 度，如图 1.10 所示。



图 1.10 输入并执行“RT”命令及效果

### 金钥匙

在已执行完的命令中再次按回车键,会让小海龟又一次执行该行命令。所以在使用这种方式执行命令时,一定要判断是否需要重复执行该行命令。



第 4 步：在新一行命令提示符“？”后输入“FD 140”，按回车键，小海龟在图形窗口中画出如图 1.11 所示的折线。

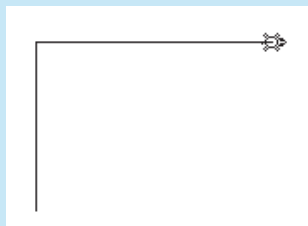


图 1.11 画完折线

### 知识窗

#### 步 长

步长的具体值就是指距离的长短(小海龟每 1 个步长的默认值为 0.5 厘米)。如“FD 100”中步长值为 100, 是让小海龟前进 50 厘米。

步长可以是具体数值, 也可以是一个算式, 小海龟会自动计算算式, 再按计算的结果完成任务。在 LOGO 语言中, 加号、减号、乘号、除号分别用“+”“-”“\*”“/”符号表示。



### 试试看

重新打开 LOGO 窗口, 输入命令, 让小海龟在图形窗口中画出如图 1.12 所示的折线图形。

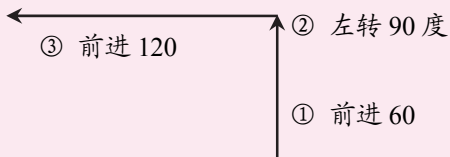


图 1.12 示意图



### 3. LOGO 语言的常用控制命令

在实际应用中, 很多情况下并不需要小海龟画出它的全部运动轨迹。例如画图 1.13 所示马路中间表示分界线的虚线。



图 1.13 马路中间分界虚线示例

能不能让小海龟在需要画画时使用它的“神奇画笔”, 不需要画画时收起它的“神奇画笔”, 画完画“回家”后“隐藏”在幕后休息, 需要时又能及时“显身”呢? 使用表 1.3 中的命令, 可以让小海龟正确地使用它的“神奇画笔”, 让它来去自如。



表 1.3 LOGO 中的基本命令 2

命 令	格 式	作 用
提笔 (Penup)	PU	提起画笔，此后小海龟移动时不再画线
落笔 (Pendown)	PD	落下画笔，此后小海龟移动时将恢复画线
隐龟 (HideTurtle)	HT	将小海龟隐藏起来，但不影响小海龟执行各种命令
显龟 (ShowTurtle)	ST	将隐藏到后台的小海龟显示出来
回家 (Home)	HOME	小海龟回到“母位”，头向上待命



## 做一做

参照图 1.13 所示，让小海龟画出马路中间的分界线。要求：虚线部分长为 90cm，实线部分长为 30cm；虚线部分每个小线段长 10cm，各小线段之间相距 5cm，画完后为了避免小海龟留在画面上影响美观，让它“回家”后隐身。

第 1 步：启动 LOGO，在听者窗口的命令提示符“？”后输入命令“RT 90”，按回车键，小海龟在图形窗口“母位”处向右旋转 90 度，如图 1.14 所示。

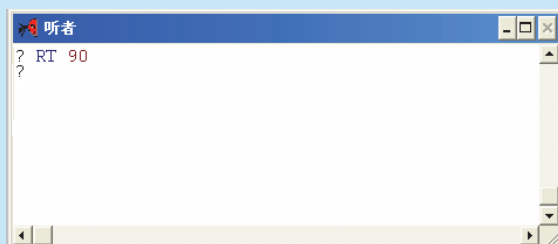


图 1.14 让小海龟右转 90 度

## 小博士

如果在刚执行完 LOGO 命令的窗口中开始输入新的命令，通常需要对 LOGO 环境进行“初始化” (Draw)，清除系统中遗留的数据 (听者窗口中的命令和图形窗口中的图形)，让小海龟回到“母位”，做好开始工作的准备。“初始化”命令如表 1.4 所示。

表 1.4 “初始化”命令

命 令	格 式	作 用
初始化 (Draw)	DRAW	使 LOGO 系统处于刚启动后的初始状态

第 2 步：在新一行命令提示符“？”后输入命令“FD 20”，按回车键，小海龟在图形窗口中前进 10cm 并画出线段，如图 1.15 所示。



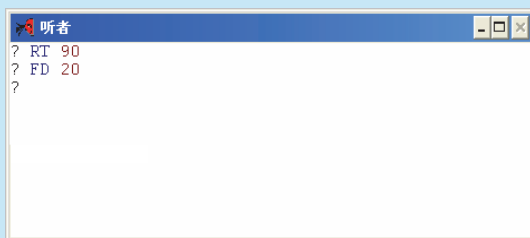


图 1.15 画出虚线中的第一个实线段

### 小博士

在听者窗口中输入命令时，可以通过颜色来判断其正误：当输入的命令及格式正确时，命令名称用蓝色显示，参数用红色显示，命令中如果有中括号，则用粉红色显示，如图 1.16 所示；当输入的命令有误时，该命令行会变为墨绿色。

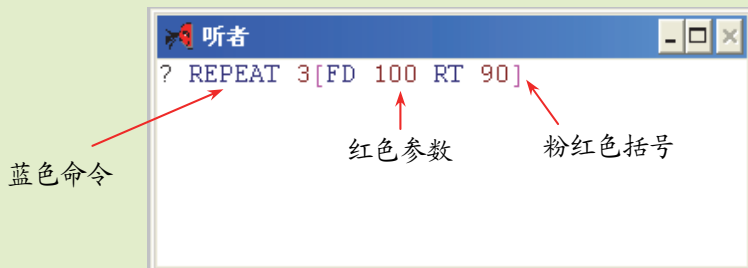


图 1.16 命令行中的提示颜色

所以，我们一旦输入错了命令，该命令行会变为墨绿色，如果按回车键执行这条命令，系统会给出发生错误的提示。例如：如果把“FD 100”输成了“FD100”，按回车键后会出现如图 1.17 所示的情况，指出输入的命令有误。

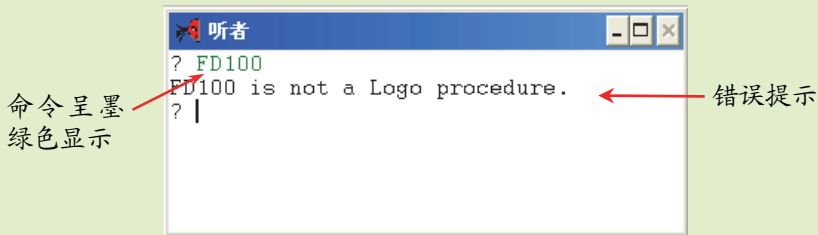


图 1.17 输错命令后，听者窗口中给出发生错误的提示

在听者窗口中修改命令，与在“记事本”程序中修改字符的方法一样：单击出现错误的地方，将插入点光标移到命令行中 D 的错误字符处，直接修改即可。



第3步：在新一行命令提示符“？”后依次输入命令“PU FD 10”，按回车键，小海龟在图形窗口中向前“跳跃”5cm，如图 1.18 所示。



图 1.18 小海龟不画线向前“跳跃”

### 金钥匙

如果需要小海龟连续做多个动作，可以在听者窗口的同一行中输入多条命令后再按回车键，小海龟将依次执行这几条命令。在同一行中输入多条命令时，各条命令之间必须用空格隔开。



第4步：在新一行命令提示符“？”后依次输入命令：“PD FD 20 PU FD 10 PD FD 20 PU FD 10 PD FD 20 PU FD 10 PD FD 20 PU FD 10 PD FD 20 PU FD 10 PD FD 60”，按回车键，小海龟在图形窗口中画完剩余的虚线和实线，如图 1.19 所示。

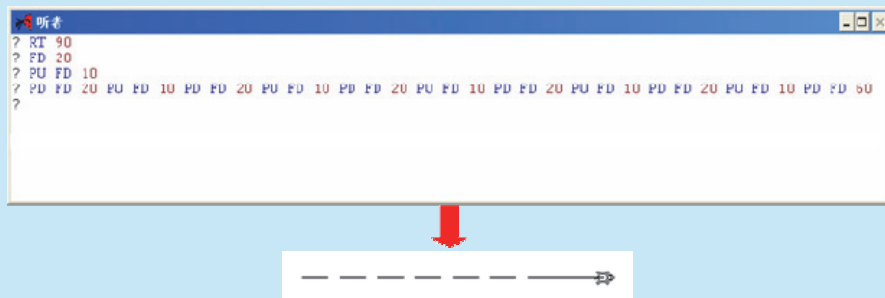


图 1.19 小海龟画出完整的马路中间的分界线

第5步：在新一行命令提示符“？”后依次输入命令：“PU HOME”，按回车键，小海龟抬笔“回家”，如图 1.20 所示。

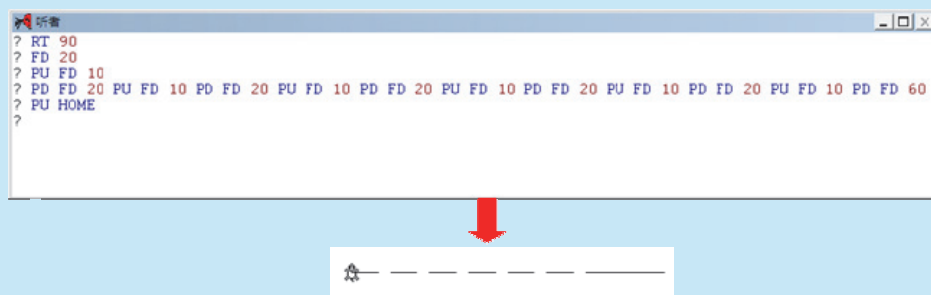
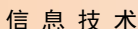


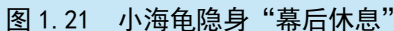
图 1.20 小海龟抬笔“回家”



执行“回家”命令后，无论小海龟身处“图形”窗口中的何处，它都会直接回到“母位”，头向上待命。在这个过程中，如果当前小海龟处于“提笔”状态，就不再画线直接回到“母位”；如果小海龟处于“落笔”状态，还会画出一段以当前位置为起点、“母位”为终点的直线。可以利用 HOME 命令的这一特点画图，非常方便。

表 1.5 两种画等边三角形的方法

第6步：在新一行命令提示符“?”后输入命令“HT”，按回车键，小海龟“隐身”到幕后休息，这样一个完整的马路中间的分界线呈现在图形窗口中，如图1.21所示。



## 擦除和清屏

1. 图 1.22 所示是王刚同学在指挥小海龟画完“回”字后,忘了让它“提笔”就“回家”了的屏幕截图!



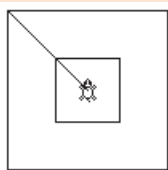


图 1.22 王刚同学画“回”字的屏幕截图

怎么解决这个问题呢？是不是让小海龟执行 DRAW 命令后从头再来？不用，那样既浪费时间还可能再次出错！LOGO 为小海龟准备了一个修正错误的命令——“笔擦”（PenErase）命令，如表 1.6 所示。

表 1.6 “笔擦”命令

命 令	格 式	作 用
笔擦 (PenErase)	PE	擦除小海龟移动时经过位置上已画出的图形

执行“笔擦”（PE）命令后，小海龟就变成了“橡皮擦”，它的作用和“画图”程序中的“橡皮擦”作用相似，利用它不仅可以修改画错的地方，还可以作为特殊的“画笔”画画呢！例如，与“前进”命令、“后退”命令结合使用，可以画虚线：让小海龟先前进画一段实线，然后使用“笔擦”和“后退”命令擦掉不需要的部分。

2. 在画新图形时，通常需要在听者窗口中输入“清屏”（ClearScreen）命令，擦除图形窗口中以前画过的所有图形，同时让小海龟回到母位且头向上，如表 1.7 所示。

表 1.7 “清屏”命令

命 令	格 式	作 用
清屏 (ClearScreen)	CS	清空图形窗口，小海龟回到母位，头向上待命

3. 当听者窗口中的命令行太多、不利于阅读输入的信息时，就要用到“清空听者窗口”命令 CT，如表 1.8 所示。

表 1.8 “清空听者窗口”命令

命 令	格 式	作 用
清空听者窗口 (CT)	CT	清空听者窗口中的所有内容



### 试试看

使用“显龟”命令和“清空听者窗口”命令，让小海龟画出图 1.23 所示的图形，再隐身。试着在一行中输入完所有命令。

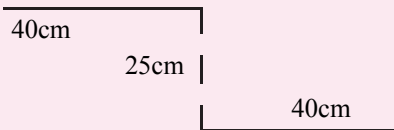
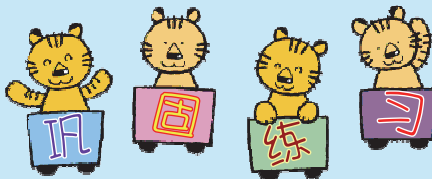


图 1.23 示意图



### 说说看

1. 在上面的“试试看”中，指挥小海龟画图前不使用“显龟”命令而使用“初始化”命令可以吗？
2. “清屏”“初始化”和“清空听者窗口”命令的作用有什么不同？
3. 在 LOGO 语言中，除 HOME 命令外，能使小海龟“回家”的命令有哪些？它们与 HOME 的作用有什么不同？



1. 判断下列说法是否正确。
  - ① 在“图形+听者窗口”中，可以同时激活图形窗口和听者窗口。( )
  - ② 同“画图”程序一样，LOGO 也是一个 Windows 应用程序。( )
  - ③ CS、HOME、DRAW 命令都可以让小海龟回到母位。( )
  - ④ 命令“FD 100 BK 100”和“FD 100 HOME”的执行结果一样。( )
  - ⑤ 执行了“隐龟”命令后，小海龟就不能继续画图了。( )
2. 启动 LOGO 系统后，在默认状态下，下列哪个选项不在窗口中显示。( )
  - A、工具栏
  - B、编辑窗口
  - C、图形窗口
  - D、听者窗口



3. 可以将 LOGO 窗口切换到全屏幕图形显示方式的快捷键是( )。

A、Shift + F2

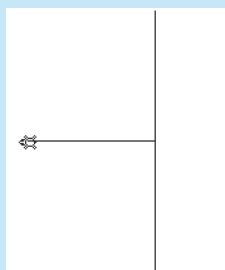
B、Shift + F3

C、Shift + F4

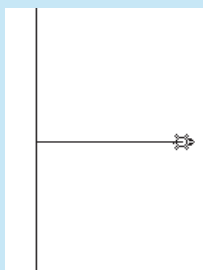
D、Shift + F5

4. 在听者窗口中输入如下命令，处在初始状态的小海龟画出的图形是( )。

FD 200 RT 180 FD 100 RT 90 FD 100



A



B



C



D

5. 在下列关于 HOME 命令的说法中，错误的是( )。

A、小海龟在回归母位时可以画出一条线段

B、小海龟在回归母位时可以不画线段

C、小海龟在回归母位时一定能画线段

D、小海龟在回归母位时，既可以画出线段，又可以不画线段，这取决于它是否处在“落笔”状态

6. 比较使用 LOGO 程序作图与使用“画图”软件作图的方式有什么不同？

7. 简述 LOGO 三种窗口的不同作用 and 特点。

8. 指挥小海龟画出如图 1.24 所示的图形。

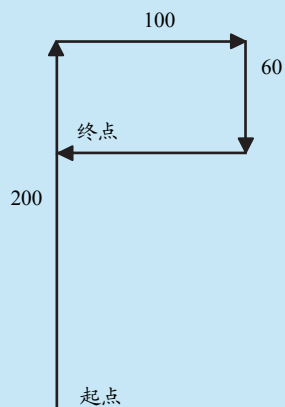
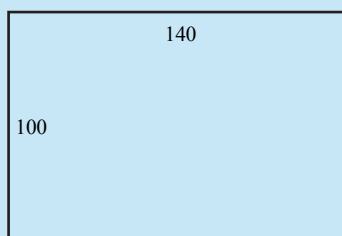


图 1.24 矩形和旗帜



9. 指挥小海龟画出如图 1.25 所示的飞机图形。

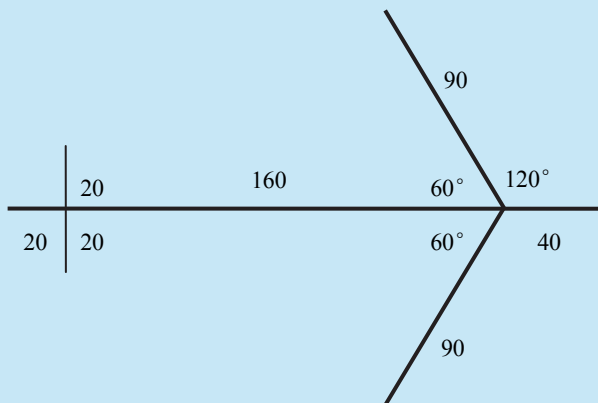


图 1.25 飞机图形

10. 指挥小海龟画出图 1.13 所示的马路示意全图。



## 实践活动

小海龟已经学会画线条了。现在，让小海龟参照图 1.26 所示“我的绿色家园”设计图，先“修”路、再“造”山，画出设计图纸。

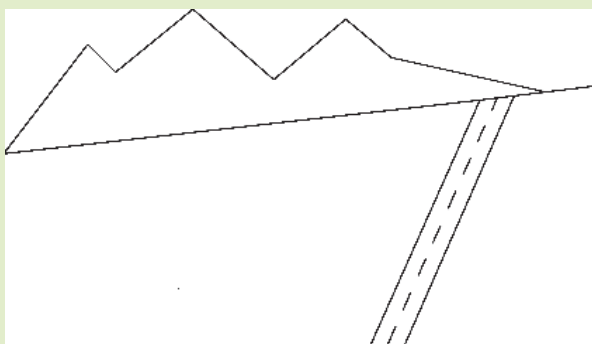


图 1.26 “我的绿色家园”设计图



## 阅读材料

## LOGO 语言简介

LOGO 语言是一种与自然语言非常接近的编程语言，它通过“绘图”的方式来学习编程，适合儿童的认知特点，能充分引起他们的兴趣和积极性，达到寓教于乐的目的，是对初学者特别是儿童学习程序设计语言的最佳选择。

LOGO 语言创始于 1968 年，是美国国家科学基金会所资助的一项专案研究，在麻省理工学院(mit)的人工智能研究室完成。logo 源自希腊文，原意为思想，最初的 LOGO 语言是由一名叫佩伯特的心理学家设计的。他在从事人工智能的研究时，一个像海龟的机械装置，触发了他的灵感。佩伯特希望孩子不要机械地记忆事实，强调创造性的探索。他说：“学校的多数课程是记忆一些数据和科学事实，却很少着眼于真正意义上的学习与思考。”他利用广博的知识及聪明的才智完成了 LOGO 语言的设计。

LOGO 语言是一种结构化程序设计语言。它是交互式的，为人们提供了良好的编程环境；它是模块化的，便于程序的修改与扩充；它是过程化的，包含了过程、参数、变量等重要概念，并允许递归调用；它有丰富的数据结构类型；有生动的图形处理功能；它有低年级学生容易掌握的基本绘图指令，在此基础上，高年级学生学 LOGO 程序设计自然不是难事。

LOGO 语言的功能不止于此，它拥有许多语言不可替代的作用，比如强大的计算能力、先进的视窗理念，LOGO 语言的许多理念是可以代表人类认识的规律的，学习这些，可以让人变聪明。

LOGO 语言从开始发展到现在，已有多种版本，从上世纪 90 年代起就有了中文版的 LOGO 语言，如“易乐谷/ELOGO”、“创新 CX-LOGO/微世界”和“Flash 版 Logo 语言”。

参考资料：<http://baike.baidu.com/>





## 第2单元 LOGO 的过程



在第1单元我们指挥小海龟画图的时候，总是需要一行行地输入命令。这样虽然很直观，但也有一个很大的缺点，那就是所有的命令都没保存下来，如果需要再画一次，只好重新输入，非常麻烦。为了解决这一问题，我们可以把画一个图形的所有命令组合在一起，组成一个“过程”，给它取一个名字，并且让小海龟记住这个“过程”，以后只要输入这个名字，小海龟就可以凭“记忆”调用这个“过程”完成任务。

### 一、过程的定义和保存

在 LOGO 中，把小海龟完成任务时所用的命令按一定的规则放在一起，给它取个名保存起来，就形成了一个过程。过程可以保存、调用、修改。

LOGO 语言中的过程要按一定的规则编写。一个完整的过程由三部分组成：

过程头——TO 过程名

过程体——完成工作的命令组

过程尾——END

过程头：过程开始的标志，以 TO 开头，后面是过程名，中间用空格隔开。过程名一般是英文字母、符号或数字的组合。过程名不能和 LOGO 语言中已有的命令同名，也不能包含“+”“-”“\*”“/”等运算符和空格。过程名最好采用与过程的功能及特点有联系的英文单词(或英文单词缩写)，也可以采用汉语的拼音字母缩写。

过程体：完成某个任务的若干条 LOGO 命令的集合。

过程尾：END，用来表示一个过程的结束。

一个过程的产生要经历编写、定义、运行和保存这一系列环节。

编写过程需要使用 LOGO 的“编辑”(Edit)命令打开“编辑”窗口进行。“编辑”命令的具体内容如表 2.1 所示。

表 2.1 “编辑”命令

命 令	格 式	作 用
编辑	EDIT [过程名]	打开并进入编辑窗口

在听者窗口输入“EDIT 过程名”(这里以“EDIT LX”为例)，按回车键，就会打开如图 2.1 所示的编辑窗口，此后就可以在闪动的插入点光标处依次输入过程体包含的一系列命令，完成过程的编写。格式中的方括号表示过程名可以省略。

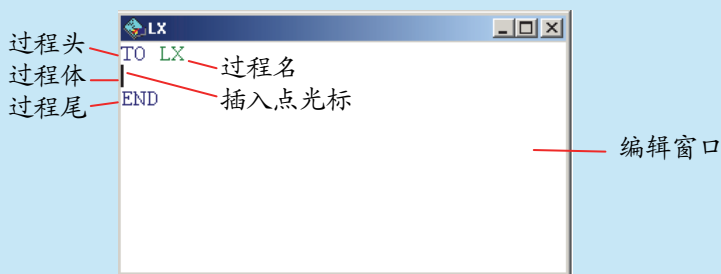


图 2.1 执行“EDIT LX”命令后打开“LX”过程编辑窗口

如果输入“编辑”命令时省略了过程名(只输入“EDIT”),将打开一个空的编辑窗口,如图 2.2 所示。

这时,需要先输入过程头(TO 过程名),然后输入过程体——命令集合,过程体输入完之后按回车键,另起一行输入过程尾“END”。

要使用编制的过程,只要在听者窗口输入过程的名称并按回车键,小海龟就会执行该过程中的所有命令,完成过程对应的任务。




图 2.2 空的编辑窗口



编写一个名为“ZFX”的过程,调用该过程时可以画出边长为 50cm 的正方形,运行无误后,以“ZFX.LGO”为文件,保存到“D:\六(2)班”文件夹中自己的文件夹里。

第 1 步:启动 LOGO,在听者窗口命令提示符“?”后输入“EDIT ZFX”,打开编辑窗口。

### 金钥匙

也可以在听者窗口命令提示符“?”后输入“EDIT”或直接单击工具栏中的“编辑”(Edit)工具图标,打开空编辑窗口,在插入点光标处输入过程头“TO ZFX”后按回车键,如图 2.3 所示,然后开始输入过程体的内容。

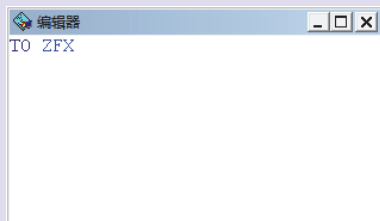


图 2.3 在空编辑窗口中输入过程头





第2步：在插入点光标处输入画正方形的命令，如图2.4所示。

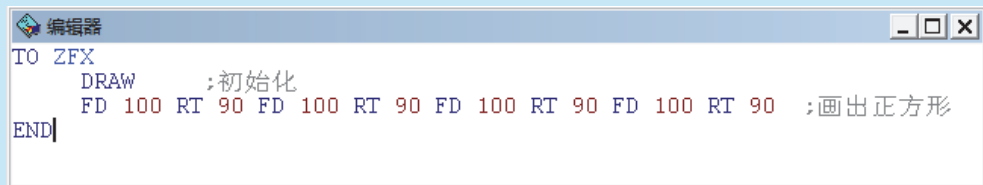


图 2.4 输入过程体内容

### 小博士

1. 在输入一些复杂的代码时，可以在一行或一段命令行的末尾输入一个空格后再输入一个英文分号“;”，接着输入一些文字以解释命令的功能和作用。“;”后面的内容称为“注释语句”，注释语句不执行，仅起到解释说明的作用。例如第2步命令行中的“;画出正方形”就是注释语句。

2. 为了突出过程的层次结构，过程体中的命令应采用缩进格式——一般情况下每个层次缩进一个制表位。具体操作为：将插入点光标移到要缩进进行的最前端，按一下 **Tab** 键。



第3步：检查输入的“ZFX”过程体的内容，确认没有错误后按 **F2** 键，编辑窗口最小化，返回听者窗口，出现一个命令行“ZFX defined”，表示小海龟已经记住了“ZFX”过程内容，如图2.5所示。

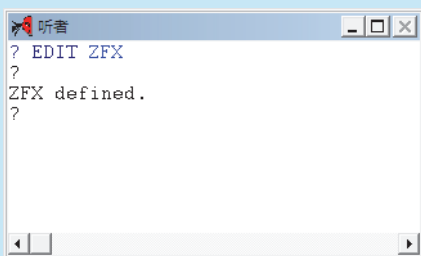


图 2.5 按定义过程键 **F2** 后的听者窗口

### 小博士

在编辑窗口中编写完一个“过程”后，小海龟还不能立即记住“过程”的内容，要按 **F2** 键“提醒”小海龟一下噢！

按 **F2** 键让小海龟记住“过程”内容的操作，称为“定义”过程。



第4步：在听者窗口的命令提示符“?”后输入过程名“ZFX”，按回车键，小海龟画出正方形，如图2.6所示。



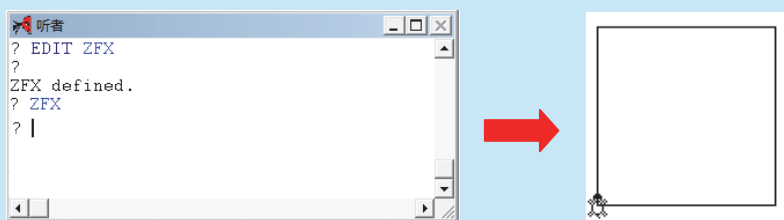


图 2.6 调用“ZFX”过程及运行结果

第 5 步：打开“窗口”菜单(如图 2.7 所示)，单击“7 ZFX”，使“ZFX”过程的编辑窗口成为当前窗口；然后打开“文件”菜单，执行保存(S)命令，打开“保存 PC Logo 文件”对话框(如图 2.8 所示)，选择 D 盘“六(2)班”文件夹中自己的文件夹(这里以“李丹宁”文件夹为例)，在“文件名(N):”框中输入“ZFX.LGO”后单击 **确定** 按钮。

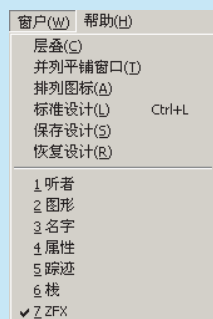
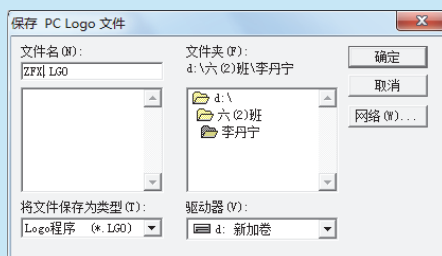


图 2.7 激活“ZFX”过程编辑窗口



图 2.8 执行“文件”菜单中的“保存”命令



### 小博士

在编辑窗口中定义了过程后，在听者窗口中输入过程名运行它，如果运行效果不满足我们的要求，还可以回到编辑窗口对它进行修改。

如果对已保存过的过程进行修改后，没有再执行保存操作，在关闭编辑窗口前会弹出图 2.9 所示的对话框，提醒你是否需要保存修改后的过程。此时，单击 **是(Y)** 按钮，保存修改后的过程；单击 **否(N)** 按钮，不保存修改后的过程；单击 **取消** 按钮，放弃关闭编辑窗口的操作。

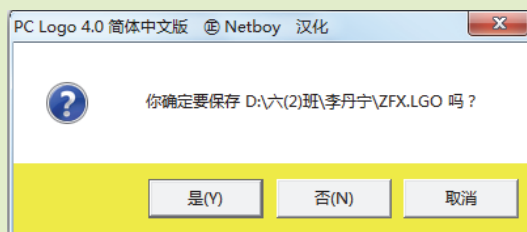


图 2.9 提醒保存过程的对话框

第 6 步：关闭编辑窗口，退出 LOGO 系统。





## 说说看

使用“EDIT”命令时，带参数和不带参数有什么不同？



## 知识窗

## 使用 LOGO 命令保存过程文件

保存过程文件有两种方法，一种是我们熟悉的使用“文件”菜单命令，另一种是使用 LOGO 命令。例如，在编辑窗口中定义了过程后，可以在听者窗口中使用“存盘”(SAVE)命令进行保存。“存盘”命令的具体内容如表 2.2 所示。

表 2.2 “存盘”命令

命 令	格 式	作 用
程序存盘 (SAVE)	SAVE "文件名	将工作区中的所有过程存入指定磁盘中

使用此命令时，命令中的文件名可以省略扩展名，文件名前的英文双引号与文件名间没有空格。

例如要把工作区中的过程保存为 SJX6 文件，在听者窗口命令提示符后输入：

SAVE "SJX6

按回车键后，听者窗口出现以下提示信息：

Saving workspace into file SJX6.LGO

Result: TRUE

表示已经把工作区中的过程用 SJX6.LGO 为名保存在 LOGO 默认的“C:\Program Files\PC Logo 4.0”文件夹中。



## 试试看

1. 编写一个画正五边形的过程，以“ZWBX.LGO”为文件名保存在自己的文件夹中。
2. 保存好的过程文件，在重新启动 LOGO 后还可以再装载进来使用。请试着使用“文件”菜单中的“装载”命令，装载此前保存好的 ZFX 过程。



## 二、使用重复命令

小海龟学会了使用“过程”的新方法，工作效率提高了许多哟！但我们也发现，在小海龟前面创作的一些作品中，画同一图形的命令中有许多命令是重复的。例如：画一个边长为 100 的正方形命令是：

```
FD 100 RT 90 FD 100 RT 90 FD 100 RT 90 FD 100 RT 90
```

从这组命令中可以看出，画这个正方形的基本命令只有两条，即“FD 100”和“RT 90”，只不过是把它重复使用了 4 次。

正三角形(等边三角形)、正方形、正五边形、正六边形……正 N 边形等，它们的特点是各边相等、各角也相等，这种多边形称为正多边形。

上面的这个例子是一个典型的画正多边形问题。按上述方法处理这类问题，工作效率还是较低的。在 LOGO 语言中，使用“重复”(REPEAT)命令可以简化多次连续重复使用的命令序列(一个或多个命令)。“重复”命令的具体内容如表 2.3 所示。

表 2.3 “重复”命令

命 令	格 式	作 用
重复(REPEAT)	REPEAT 重复次数[命令序列]	按指定次数重复执行所规定的命令序列

重复命令必须和其他命令一起使用，命令中的重复次数是自然数，若输入小数，小海龟将忽略小数部分；“命令序列”必须用英文方括号括起来，重复次数和方括号之间没有空格。

如果使用重复命令画一个边长为 100 的正方形，执行的命令应该是：

```
REPEAT 4[FD 100 RT 90]
```

显然，重复命令简化了多次连续重复执行的命令序列，使表达变得更加简洁紧凑。



画出边长为 5cm 的正三角形、正五边形、正六边形、正八边形、正三十六边形。

第 1 步：启动 LOGO，在听者窗口命令提示符“？”后输入“EDIT ZDBX”，按回车键进入编辑窗口。

第 2 步：在插入点光标处输入以下过程体：

```
DRAW
```

```
REPEAT 3[FD 10 RT 120]
```





### 小博士

由于重复命令不能单独使用，因此，在使用重复命令前，需要先针对基本问题设计命令序列，然后再根据需要重复的次数使用重复命令。



检查确认无误后按 **F2** 键，编辑窗口最小化，听者窗口出现 “ZDBX defined”，在命令提示符 “?” 后输入 “ZDBX”，按回车键，小海龟画出一个正三角形。

### 小博士

小海龟画出正三角形的一条边后，需要转 120 度再画下一条边，画一个完整的正三角形转过  $3 \times 120$  度，正好转一周 (360 度)。同理，画出正四边形的一条边后要转 90 度，画一个完整的正四边形转过  $4 \times 90$  度；画一个完整的正五边形转过  $5 \times 72$  度…… 画一个完整的正 N 边形转过  $(N \times 360/N)$  度。

由以上分析我们发现，正多边形边数就是画正多边形时的重复次数，它与每次转过的角度的乘积总是 360 度，画正 N 边形的每条边后转过的角度是  $360 \div N$ ，那么，画正 N 边形的命令可以写为：

REPEAT N[FD 边长 RT  $360/N$ ] ;N 为边数



第 3 步：打开 “窗口” 菜单，单击 “ZDBX” 过程名，使 “ZDBX” 过程的编辑窗口成为当前窗口，在过程体中添加画正五边形、正六边形、正八边形、正三十六边形的命令，如图 2.10 所示。

```
TO ZDBX
DRAW
REPEAT 3[FD 10 RT 120]
REPEAT 5[FD 10 RT 360/5]
REPEAT 6[FD 10 RT 360/6]
REPEAT 8[FD 10 RT 360/8]
REPEAT 36[FD 10 RT 360/36]
END
```

图 2.10 编辑 “ZDBX” 过程

### 小博士

分析以上程序可以看出，这个程序是按照由上到下的顺序一行一行地执行的，程序执行过程中没有分支、没有重复，这种结构称为顺序结构。顺序结构是一种最基本、最简单的程序结构。



第 4 步：检查输入的过程体，没有错误后按 **F2** 键，在编辑窗口最小化后，听者窗口中出现 “ZDBX redefined”，在命令提示符 “?” 后输入 “ZDBX” 后按回车键，小海龟画出正三角形、正五边形、正六边形、正八边形和正三十六边形，如图 2.11 所示。

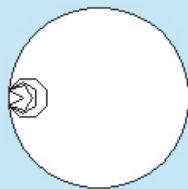


图 2.11 画正多边形

第 5 步：保存程序文件到自己的文件夹中。

### 试试看

参照图 2.12 所示，编写过程让小海龟“写”一个“米”字。要求：以“母位”为出发点，每个笔画长均为 10cm。

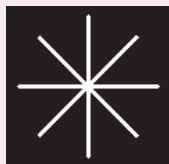


图 2.12 “米”字示意图



## 三、带参数的过程

分析以上设计“正多边形”过程我们发现，正多边形的“边长”和“边的个数”是决定小海龟画出不同正多边形的重要数据，除此以外其他数据都是不变的。这种情况是常见的：正方形大小由边长 1 个数据决定，长方形的大小由长和宽 2 个数据决定，圆的大小由半径 1 个数据决定……如果在设计过程时，能将这些重要数据设置成变量，针对这类问题的程序设计就会大大简化。

在 LOGO 语言中，过程中的变量叫做参数，含有变量的过程称为带参(数)过程。带参(数)过程的格式是：

```
TO 过程名 :参数 1 :参数 2 .....
    过程体
END
```

编写带参过程时，过程名与参数之间、参数与参数之间要用空格隔开，参数前加英文冒号“:”，且冒号与参数间不能有空格。

### 1. 带一个参数的过程

这里以画正方形为例(本例取正方形的边长为 20 步长)，通常的方法是：

① 先编写画边长为 20 步长的正方形的过程：

```
TO ZFX
    REPEAT 4[FD 20 RT 90] HOME
END
```

② 定义这个过程，然后执行它。

这里只是画一种边长的正方形，我们不会感觉到困难，但是如果画任意边长的正方形，用这个方法就很麻烦。如果使用带参过程，问题就简单多了。

我们知道，画正方形只有一个变量，就是它的边长，如果将上面的过程设计为一个带参过程：





```
TO ZFX :A  
  REPEAT 4[FD :A RT 90] HOME  
END
```

定义后在听者窗口输入“过程名 参数值”，执行这个过程就可以画出任意边长的正方形。

### 试试看

画出边长分别为 30、40、50、60、70、80 步长的正方形。




## 2. 带多个参数的过程

用画正方形的方法画任意正多边形，不仅要涉及“边长”，还要涉及“边的个数”，这就需要设计带多个参数的过程。



### 做一做

编写一个可以画任意大小正多边形的过程，调试无误后以“ZDBX\_C.LGO”为名保存该过程，并使用这个过程画出边长为 8 的正三角形、边长为 9 的正五边形、边长为 10 的正八边形……

第 1 步：启动 LOGO，单击工具栏中的“编辑(Edit)”图标，打开“编辑”窗口。

第 2 步：输入“TO ZDBX\_C :N :A”（“:N”代表正多边形边的个数，“:A”代表正多边形的边长），如图 2.13 所示。

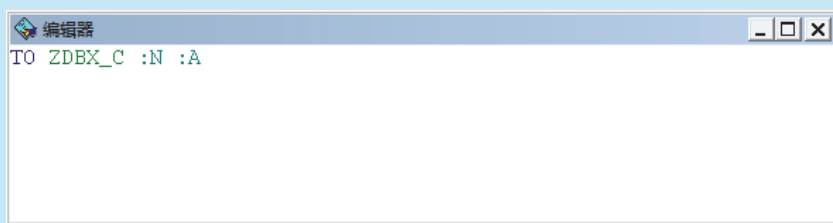


图 2.13 带参数的过程头

第 3 步：按回车键，在插入点光标处将过程补充完整，如图 2.14 所示。

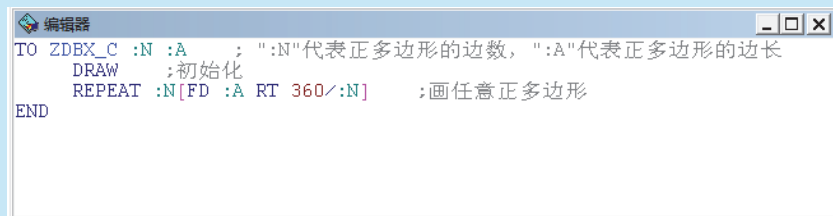


图 2.14 画任意正多边形的过程

第4步：检查输入的带参过程，没有错误后按 **F2** 键，在编辑窗口最小化后听者窗口中出现“ZDBX\_C defined”。

第5步：在听者窗口命令提示符“?”后依次输入：

“ZDBX\_C 3 8”，按回车键；

“ZDBX\_C 5 9”，按回车键；

“ZDBX\_C 8 10”，按回车键；

小海龟分别画出正三角形、正五边形和正八边形。

### 小博士

设计具有共同属性的带参过程的好处在于提高过程的通用性。



第6步：单击听者窗口的标题栏，然后打开“文件”菜单，执行另存为(A)...命令，在打开的“保存 PC Logo 文件”对话框中，以“ZDBX\_C.LGO”为文件名将程序保存在自己的文件夹中。

### 知识窗

#### 画几何图形

前面我们让小海龟画过的三角形、长方形、正方形、正多边形都是常见的几何图形。除此之外，我们熟悉的几何图形还有梯形、扇形、圆等。在这些常见的几何图形中，三角形、梯形利用画线条就能快速画出；长方形、正方形、正多边形及圆的形状都由一个或几个变量决定，我们可以找出它们的规律。

由画长方形命令“REPEAT 2[FD :a RT 90 FD :b RT 90]”可知，只要我们给表示长的变量 a、表示宽的变量 b 赋值，就可以画出任意大小的长方形和正方形。同样，利用“REPEAT :N[FD 边长 RT 360/:N]”命令，给 N 赋值后，就可以画出任意边数的正多边形。

在画正多边形时我们发现，当边数越多时，画出的图形就越接近圆。一般情况下，当正多边形的边数 N 大于等于 36 时，画出的正多边形近似于圆。由此可以通过计算出小海龟画边数为 36 的正多边形每次移动的步数，编写出画圆的命令。

当 N=36 时，小海龟画一个完整的正三十六边形移动的距离近似等于圆的周长  $2\pi r$ ，则画每一条边的步长为： $2\pi r \div 36 = 2 \times 3.1416r \div 36 \approx 0.175r$ 。

由此可得，用画正多边形近似画半径为 r 的圆的命令为：

```
REPEAT 36[FD 0.175*:r RT 360/36]
```







LOGO 语言中还另外给出了“画矩形”(又叫“长方形”)命令和“画椭圆”命令(正方形和圆分别是长方形和椭圆的特例),这两个命令的具体内容见表 2.4。

表 2.4 “画矩形”和“画椭圆”命令

命 令	格 式	作 用
画矩形 (STAMPRECT)	STAMPRECT 长 宽	画空心矩形
	(STAMPRECT 长 宽 "TRUE)	画实心矩形
画椭圆 (STAMPOVAL)	STAMPOVAL 长半轴 短半轴	画空心椭圆
	(STAMPOVAL 长半轴 短半轴 "TRUE)	画实心椭圆

参数中用到的符号“”“(”“)必须是英文符号,参数与参数之间用空格隔开。使用这两个命令不能画出与水平方向有任何倾斜的矩形和椭圆。

使用 STAMPRECT 命令画矩形时,以小海龟所在的位置为矩形左下角向右上方伸展,如图 2.15 所示,而与小海龟的当前方向无关。当给出的长、宽数据相同时,画出的是正方形。

如图 2.16 所示,STAMPOVAL 命令中的长半轴是  $OA_1$  或  $OA_2$  的值,短半轴是  $OB_1$  或  $OB_2$  的值,画椭圆时,海龟位于椭圆的中心  $O$  点处,当长半轴等于短半轴时,画出的是圆。

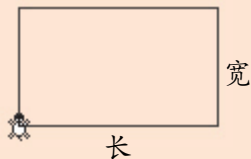


图 2.15 画矩形

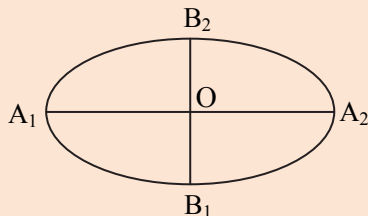


图 2.16 椭圆示意图

### 试试看

圆弧是圆的一部分,常用到 60、90、120、180 度的圆弧,也就是六分之一、四分之一、三分之一圆及半圆。

由前面使用的画圆命令“REPEAT 36[FD 0.175\*r RT 360/36]”可知,重复数 36 对应全圆,那么画六分之一、四分之一、三分之一圆及半圆对应的重复数分别为 6、9、12、18。

依据以上数据,设计一个带参过程,让小海龟分别画出 60、90、120、180 度的圆弧。







## 四、调用过程

在 LOGO 语言中，一个过程可以调用其他过程，称之为“过程调用”；如果被调用的过程就是它本身，这种过程调用称为“递归”，这个过程就是递归过程。

在递归过程中，如果只在过程体的最后一句出现本过程的过程名，则称为“尾递归”。由于“尾递归”是自身调用，执行次数是无限的，小海龟在它的驱动下会不停地重复执行任务，直到出现图 2.17 所示的提示信息。

如果单击 ，小海龟又继续行动起来！这时，可以按  键强行终止程序运行。

使用重复、赋值命令，已经可以让小海龟又快又好地画出变化多端的图形，如果再使用递归，那会出现什么结果呢？

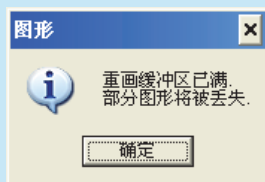



图 2.17 调用“尾递归”过程出现的提示信息



还记得图 1.1 所示的螺旋图形吗？这个图形由 50 个圆组成，最小圆的半径为 0.5 步长，相邻两个圆的半径相差 1 步长，转角相差 6 度。现在我们可以让小海龟画它了，快动手吧！

第 1 步：启动 LOGO，单击工具栏中的“编辑(Edit)”图标 ，打开编辑窗口。

第 2 步：输入“LXTX :R”过程内容。

TO LXTX :R ;R 表示圆的半径

SETBG 9 SETPC 14 ;设置背景为绿色，画笔为黄色

REPEAT 36[FD 0.175\*:R RT 360/36] RT 6

;利用重复命令画圆后再向右旋转 6 度

LXTX :R+1 ;半径累加 1 后调用自身过程

END

### 金钥匙

这个过程也可以利用“赋值”和“重复”命令实现：

TO LXTX2

SETBG 9 SETPC 14

MAKE "R 0.5

REPEAT 50[REPEAT 36[FD 0.175\*:R RT 360/36] MAKE "R :R+1 RT 6]

END





## 小博士

LOGO 语言为小海龟提供了一组画彩色图形的命令：“龟笔颜色” (SETPC)、“龟笔粗细” (SETWIDTH)、“背景颜色” (SET BACKGROUND)、“填充颜色” (FILL)。小海龟执行这些命令，就可以拿起各种颜色和粗细的画笔，使用“颜料盒”中的各种颜色，画出五彩缤纷的图画了。以上命令的具体内容参见表 2.5。

表 2.5 LOGO 画彩色图形的命令

命 令	格 式	作 用
龟笔颜色 (SETPC)	SETPC 颜色值	设置小海龟的画笔颜色。颜色的取值范围是 0~15，一个数值对应一种颜色，共有 16 种颜色，默认颜色是值为 0 的黑色
龟笔粗细 (SETWIDTH)	SETW 粗细值	设置小海龟画笔的粗细。笔粗细的取值范围是 1~999，值越大，画出的线段越粗，默认值为 1
背景颜色 (SET BACKGROUND)	SETBG 颜色值	设置图形窗口的背景颜色。颜色的取值范围是 0~15，一个数值对应一种颜色，共有 16 种颜色，默认颜色是值为 15 的白色
填充颜色 (FILL)	FILL	给封闭区域填充与其边线颜色相同的颜色。填充颜色时一定要在落笔状态下

在 LOGO 中可以直接使用公式，公式中的变量可以通过赋值命令 (MAKE) 将计算对象的值给予它。赋值命令的具体内容如表 2.6 所示。

表 2.6 赋值命令

命 令	格 式	作 用
赋值 (MAKE)	MAKE "变量名 对象	将对象的值赋给变量

赋值命令的变量名前面要添加英文符号“”，对象可以是一个运算式，也可以是用英文方括号“[ ]”括起来的字符串。对象如果是运算式，LOGO 会先计算运算式的值，再将值赋给变量。



第 3 步：检查输入的带参过程体，没有错误后按 **F2** 键，在编辑窗口最小化后，听者窗口中出现“LXTX defined”，在命令提示符“？”后输入“LXTX 0.5”，按回车键。

第 4 步：小海龟画出完整的海螺螺旋图 (参见图 1.1) 后，按 **Ctrl+G** 键终止程序。

小博士

按 **Ctrl+G** 键终止递归过程运行，对观看调用过程的结果很不方便。执行“停止” (STOP) 命令可以终止过程的运行。但使用“停止”命令还需要“条件”命令配合。“停止”和“条件”命令的具体内容见表 2.7。

表 2.7 “停止”和“条件”命令

命 令	格 式	作 用
条件 (IF)	IF 条件 THEN 命令序列	如果条件成立，则执行 THEN 后面的命令序列
停止 (STOP)	STOP	停止所在过程的执行

IF 命令中的条件，就是用关系运算符大于号“>”、大于等于号“>=”、小于号“<”、小于等于号“<=”和等于号“=”将两个常量、变量或运算式连接起来。

例如，如果我们在“LXTX :R”过程体的“LXTX :R+1”语句前添加“IF :R>50 THEN STOP”命令行，执行命令“LXTX 0.5”后，当所画圆的半径大于 50 的时候，过程就会自动停止。



第 5 步：保存程序文件到自己的文件夹中。

说说看

对比用 LXTX 过程和 LXTX2 过程这两种画螺旋图的方法，谈谈使用 LOGO 命令作图的体会。



试试看

图 2.18 是用正方形组成的螺旋图，该图由一系列边长不断增加的正方形组成，它的最小边长是 12，相邻两个边长相差 3。请编写过程指挥小海龟画出这个图形。

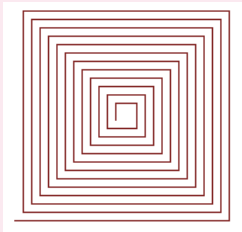
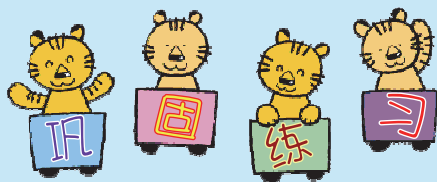


图 2.18 螺旋图形





1. 判断下列说法是否正确。

① 使用 EDIT 命令不仅能打开一个空的编辑窗口，而且还能打开正在被调用的过程的编辑窗口。( )

② 过程文件就是程序文件。( )

③ 过程只能在编辑窗口中编写。( )

④ 命令“STAMPRECT 100 50”和命令“REPEAT 2[FD 50 RT 90 FD 100 RT 90]”的执行结果一样。( )

⑤ 在设计递归过程时，合理的方法是通过设置条件使它自动停止。( )

2. 下列关于“过程名”的说法中，错误的是( )。

A、过程名中不能使用中文字符

B、过程名不能和 LOGO 系统提供的命令同名

C、过程名中可以包含“+”“-”“\*”“/”等运算符和空格

D、过程名中的英文字母不区分大小写

3. 下列关于过程的“定义”与“调用”说法中，正确的是( )。

A、在编辑窗口中编完一个“过程”，按 **F2** 键定义后才能调用该过程

B、过程一旦被定义后可以被反复使用

C、被定义的过程修改后，不需要重新定义，可直接调用得到修改后的结果

D、在听者窗口装载的过程不用定义，可以直接被调用

4. 图形窗口被激活后，执行文件菜单中的“保存”命令，不能保存下列哪种类型的文件？( )

A、.PCX

B、.LGO

C、.BMP

D、.WMF

5. 下列关于命令“MAKE "A 3 REPEAT 48[FD :A RT 90 MAKE "A :A+3]”的说法中，错误的是( )。

A、变量 A 的初始值是 3

B、每执行一次命令“MAKE "A :A+3”，变量 A 就增加 3

C、第 3 次执行命令“FD :A RT 90 MAKE "A :A+3”后，变量 A 的值是 12

D、第 4 次执行命令“FD :A RT 90 MAKE "A :A+3”后，变量 A 的值是 12

6. 在听者窗口输入并执行下列命令：

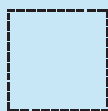


```
DRAW
REPEAT 4[FD 50 RT 90]
SETPC 2
RT 45 PU FD 5 PD FILL
```

则( )。

- A、正方形中填充了黑色
- B、正方形中填充了绿色
- C、正方形边线为黑色，中间填充了绿色
- D、正方形无法填充颜色

7. 使用命令“REPEAT 4[REPEAT 10[FD 3 PU FD 2 PD] RT 90]”可以画出图形( )。



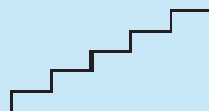
A



B



C



D

8. 指挥小海龟画出图 2.19 所示由圆组成的两个图案。

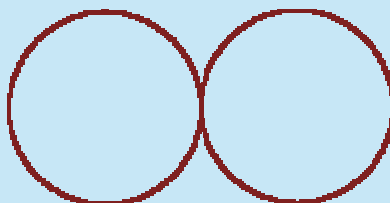
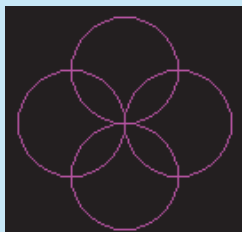


图 2.19 两个由圆组成的图案

9. 指挥小海龟画出图 2.20 所示两个由正多边形组成的螺旋图形。

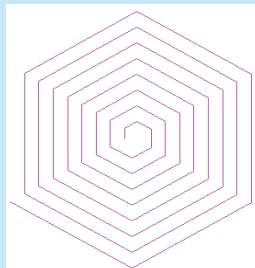
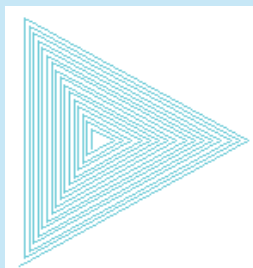


图 2.20 两个由正多边形组成的螺旋图形

10. 设计一个带参数的递归过程，指挥小海龟画出图 2.21 所示的图形。

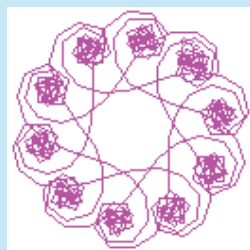


图 2.21 图形示意图



## 实践活动

让小海龟为我们画一幅《我的绿色家园》设计图。

特别提醒：小海龟完成任务后千万不要忘了保存文件！作品完成后全班同学相互展示、交流、评价。图 2.22 供参照。

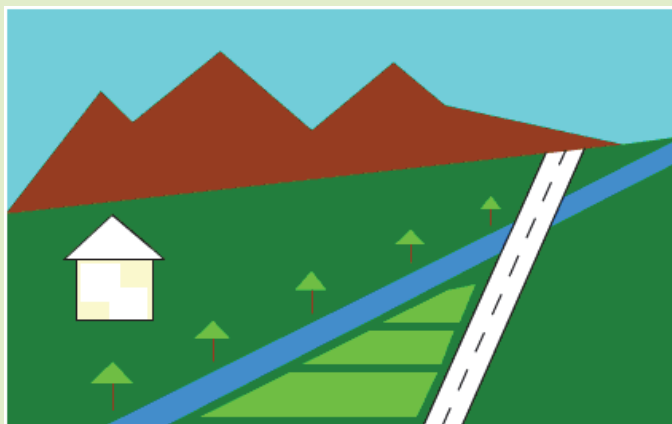


图 2.22 《我的绿色家园》示意图



## 阅读材料

## 用计算机程序解决问题

进入信息时代,用计算机帮助人们解决问题已经是我们处理日常学习、工作的基本手段。当我们用计算机解决问题时,首先要分析问题,然后根据问题的要求选择合适的软件。如果现有的软件能满足我们的要求,则可以直接用这些软件来完成任

务,例如,用计算机写一篇文稿,我们会选择“记事本”“写字板”等软件;要进行图形处理,“画图”软件就可能满足我们的要求。

除此之外,现实生活中还有许多工作比较特殊,使用现有的软件不能很好地完成,或者由于其他方面的原因无法使用现有的软件,这就需要我们编写程序来解决问题。例如,要画 100 个同心圆,使用“画图”软件就非常麻烦,这时只有自己动手编写程序来解决问题。

我们知道,计算机虽然被称为“电脑”,可以帮助人们解决问题,但它是不会自己解决问题的,需要我们根据所要解决的问题编写计算机程序,编写程序的过程就是程序设计。其基本步骤如图 2.23 所示。

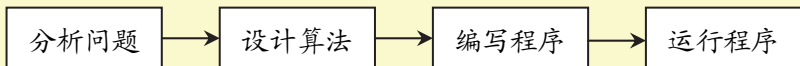


图 2.23 用计算机解决问题的过程

以 LOGO 语言程序设计为例,程序设计大致分以下几步:

① 总体构思。把要解决的问题理解透彻,按问题的要求把程序的功能描述清楚。如加工什么数据或图形,需要得到什么样的结果,用什么样的方法加工,需要使用什么样的公式,可能用到哪些工具过程等,这一步也称为设计算法。

② 编写程序。按照系统工程思想,将复杂问题的任务按功能划分为一个个独立的小任务,再为每一个小任务编写过程。这样可以将复杂的问题一层一层地化简,变成若干可直接求解的最简单子过程,而后利用过程调用将所有过程紧密联系起来,自下而上地编写、调试,完成整个程序。

③ 运行和修改。用各种有代表性的例子运行程序,确保每一个细节和每一个分支都得到测试和验证,看是否达到了预期的效果。这是对前面工作的检验,如果出现问题要查出原因,修改程序,同时也要总结经验,以纠正程序设计中出现的问题。

④ 程序优化。一个程序能够得到预想的结果,只能说明程序是正确的。一个好的程序除了正确性之外,还要做到算法先进、结构清晰、程序简洁、使用方便、适应性强等。在程序设计过程中要注意这些方面,在程序运行得出正确结果后,还要不断地改进和完善它,这对于提高程序设计能力也是非常有益的。







## 第3单元 与机器人见面



LOGO 语言中的小海龟其实就是一个生活在电脑中的机器人，我们可以用 LOGO 语言编写程序，指挥它行走、绘画、写字、唱歌、处理文字、进行数学运算…… 如果给小海龟配备合适的“装备”，让它变成现实环境中的机器人，就能用程序指挥它去灭火、走迷宫、踢足球、做游戏…… 这个设想并不是虚幻的，在后面的学习中，我们接触到的机器人就是获得了“装备”的小海龟！



### 一、初步认识机器人

机器人是一种能自动工作的机器装置，一般由执行机构、驱动装置、检测装置、控制系统和复杂机械等组成。

真正意义上的机器人应该是一个具有一定智能的计算机控制系统，它是人类制造的一种高级工具，是按照人类的某种方式进行工作的一种机器。

我们每一天都在不知不觉中同各种各样的机器人打交道，家里、学校、超市……到处都少不了机器人，如图 3.1 所示。



全自动扫地机器人



智能红绿灯



车站自动售票机

图 3.1 身边的机器人

机器人主要分为两大类：工业机器人和特种机器人。

工业机器人就是面向工业领域的多关节机械手或多自由度机器人，如图 3.2 所示。





汽车装配线上的机械手



数控切割机械手



数控车床

图 3.2 现代工业机器人

用于非制造业并服务于人类的其他机器人称为特种机器人，如图 3.3 所示。



服务机器人-海宝



水下机器人



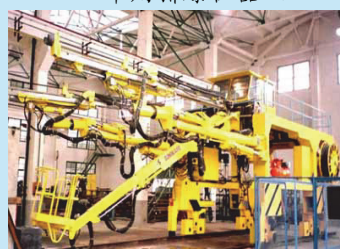
军用排爆机器人



娱乐机器人



农业机器人



隧道凿岩机器人

图 3.3 特种机器人

尽管机器人的外形、功能有千差万别的变化，但所有智能机器人的构成都可以归结为三大系统：感知系统、控制系统和执行系统。智能机器人利用其感知系统从环境获取信息，然后利用其控制系统对这些信息进行加工处理，并根据处理结果做出决定，控制执行系统去完成一定的操作。

除工业机器人和特种机器人外，还有一类为了学习机器人技术而开发的机器人，称为教学机器人。图 3.4 所示是常见的几种教学机器人。



竞赛用机器人



积木式机器人



多自由度可重构机器人

图 3.4 常见的教学用机器人





教学机器人属于结构完整、高度智能化的机器人，我们可以通过它系统学习机器人的知识。

### 说说看

你还知道哪些机器人？它们的用途是什么？



### 试试看

利用因特网查找关于机器人的知识，解决下列问题：

1. 机器人名字的由来。
2. 世界各国最出名的机器人。
3. 目前有哪些主要的机器人竞赛项目。

参考网站

<http://www.roboticfan.com/>

<http://www.kepu.com.cn/gb/technology/robot/>

<http://www.robotschina.com/>

<http://www.enet.com.cn/eschool/includes/zhuanti/zt/jqr/45.shtml>



## 二、机器人是怎样工作的

机器人的外貌有的像人，有的并不具有人的模样，但构成它们的三大系统(感知系统、控制系统和执行系统)功能相同，且与人类的感觉、大脑、运动(手和脚)系统相对应。

### 1. 感知系统

机器人通过“传感器”(见图 3.5)感知周围环境的情况。传感器很像我们人类的感觉器官，能够“看到”或“感觉到”外界环境的变化。也就是说，各种传感器就是机器人的“感觉器官”。



温度传感器

光敏传感器

声音传感器



火焰传感器

化学传感器

流体传感器

图 3.5 形形色色的机器人传感器

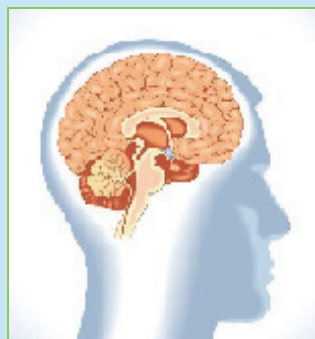
我们通常将传感器的功能与人类 5 大感觉器官相比拟：光敏传感器相当于人的视觉器官；声音传感器相当于人的听觉器官；气敏传感器相当于人的嗅觉器官；化学传感器相当于人的味觉器官；压敏、温敏、流体传感器相当于人的触觉器官。

## 2. 控制系统

机器人的控制系统相当于人的大脑，如图 3.6 所示，能够对接收到的信息进行判断，做出决定，然后控制机器人各部位协调动作。



机器人的大脑



人的大脑

图 3.6 机器人的控制系统相当于人的大脑

控制器通常由把一个计算机系统集成到一个芯片上的“单片机”承担，相当于人脑或中枢神经，具有思考和加工处理信息的功能，是机器人的指挥中心。它根据人类事先编好的程序对感知系统接受的信息进行加工、处理、决策，然后指挥执行系统工作。

## 3. 执行系统

机器人的执行系统相当于人的手、脚、肌肉、骨骼等，具有“行动”能力。

机器人的执行系统通常由机械手和移动部件组成。机械手相当于人的手，可完成各种工作；移动部件相当于人的脚，机器人靠它来“走路”，如图 3.7 所示。



图 3.7 机器人的机械手和“走路”的轮子

机器人的执行系统具备输出信息的功能，它接受控制系统的指挥，按照控制系统下达的程序命令做出相应的动作，如行走、发声、负重等。

上述三大系统与人类器官的对照关系如图 3.8 所示。

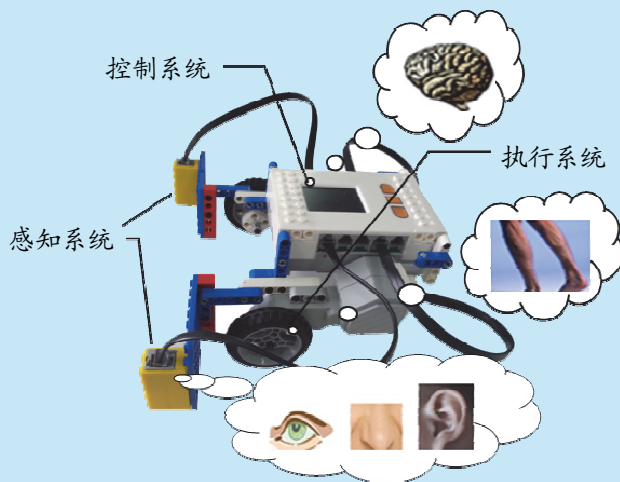


图 3.8 教学机器人的三大系统与人类的器官对比图

#### 4. 机器人的工作过程

智能机器人利用各种能够感知机器人自身或外部环境变化信息的传感器从环境获取信息，这些信息传送到机器人的控制系统，它能对接收到的信息进行判断，做出决定，然后控制机器人执行系统动作。

人与机器人沟通通常是利用电脑来实现的，人们把要求机器人完成的任务和完成任务的方法，用类似于人类的语言通过电脑编写程序(称为源程序)，然后电脑把源程序翻译为机器人的单片机能够理解的程序——机器语言程序，并把机器语言程序传送给机器人，机器人就能够根据人的要求去工作了。

#### 说说看

机器人是怎样获取、传递和处理信息的？





## 试试看

通过因特网了解机器人有哪些不同的运动方式。



## 三、机器人仿真平台

为教学机器人配套的软件系统称为机器人仿真平台。这种平台种类很多，如萝卜圈 IRobotQ3D、纳英特 NSTRSS、中鸣 3D、LEGO Digital Designer 等，它们可以在不涉及硬件的情况下，根据不同的场景搭建相应的机器人，添加各种传感器，利用计算机语言编写程序，控制机器人仿真，完成各种各样的任务。本册书使用 IRobotQ3D 作为学习机器人的仿真平台。

### 1. 安装 IRobotQ3D

IRobotQ3D 是一款在线网络软件，需要到官方网站([www.irobotq.com](http://www.irobotq.com))下载客户端，然后安装、注册、登录后才能使用。



### 做一做

老师已经把 IRobotQ3D 安装软件下载到你使用的计算机的“D:\IRobotQ”文件夹中了。请参照图 3.9 所示的步骤安装 IRobotQ3D，并注册一个你的个人账号，登录到 IRobotQ3D 在线仿真大厅，然后退出 IRobotQ3D。

第 1 步：在你使用的计算机的“D:\IRobotQ”文件夹中双击安装文件(本例为“Setup\_1.5.3.1.exe”)，按照安装向导的提示进行安装，如图 3.10 所示。



图 3.9 IRobotQ3D 客户端安装流程

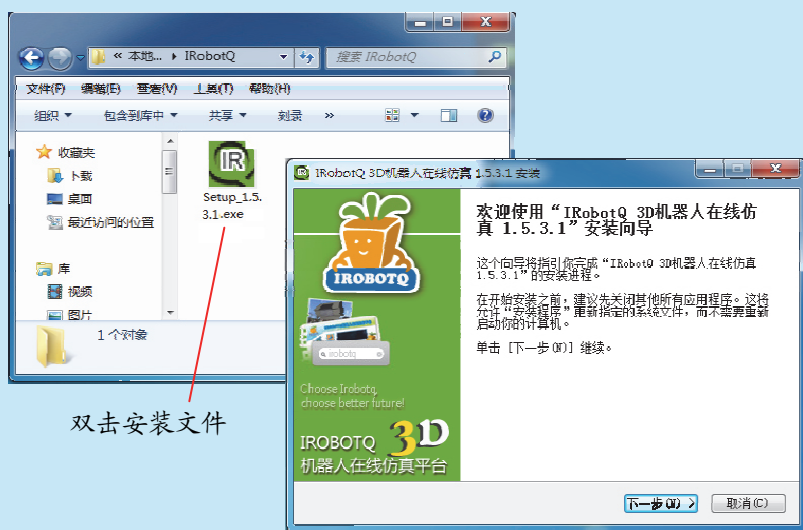


图 3.10 安装 IRobotQ3D






第2步：安装完成后，桌面上出现 IRobotQ3D 快捷启动图标。双击启动图标打开登录界面，单击 **注册**，打开注册界面，填写相关内容(本例使用“ligang”为用户名，密码为 1234)，完成注册及激活，如图 3.11 所示。



图 3.11 注册和激活账号

### 金钥匙

如果要连接到萝卜圈官方服务器，不要修改服务器的地址，保持默认地址；如果要连接到学校服务器，需要输入学校的服务器地址(请向老师索要)。



第3步：在登录界面上输入用户名和密码后单击 **Go!** 按钮，进入 IRobotQ3D 在线仿真大厅(以下简称“仿真大厅”)，如图 3.12 所示。



图 3.12 IRobotQ3D 在线仿真大厅

### 小博士

仿真大厅窗口顶部是用户信息框、工具栏和窗口控制按钮，左侧是任务列表窗格，右侧是工作区。工作区中不同阶段显示不同的内容，初始登录时显示萝卜圈官方网站内容。



第4步：单击窗口右上方的✕按钮，关闭 IRobotQ3D 窗口。

### 说说看

启动、关闭 IRobotQ3D 与启动、关闭其他软件有什么不同？



### 试试看

在仿真大厅中访问萝卜圈网站的“学习”栏目，下载萝卜圈仿真 - 入门教程，阅读第一章内容。



## 2. 认识 IRobotQ3D 机器人仿真平台

IRobotQ3D 有两种主窗口，登录后首先进入的窗口称为仿真大厅(参见图 3.12)。这个窗口除具有 Windows 应用程序窗口的一般特性外，跟现在流行的在线游戏平台很像。在这个仿真大厅中，可以在任务列表窗格中选择一项任务，单击它之后进入该任务的“房间”(如图 3.13 所示)，根据“房间”上方的“任务导航栏”提示，选择一个位置坐下，与同学一起交流、切磋机器人搭建、编程的经验，调试、修改机器人和程序，还可以与进入该房间的其他同学们比一比……当然，进行这些活动的前提是要先准备好机器人和程序。

任务导航栏



图 3.13 从“大厅”进入“工作”房间





单击“本地模式”功能选项按钮, 进入另一种主窗口——I RobotQ3D 机器人在线仿真平台, 如图 3.14 所示。



图 3.14 I RobotQ3D 机器人在线仿真平台

“I RobotQ3D 机器人在线仿真平台”（以下简称仿真平台）实际上是一个离线平台，它在本地计算机上运行，是我们搭建机器人，编写、测试程序的平台。在这个平台上，我们可以利用“构建场景”自己设置场地，让自己构建的机器人在自己编写的程序控制下，利用“开始仿真”功能进行仿真测试。



仿真平台和在线仿真大厅共同拥有“构建机器人”（在“仿真大厅”中称为“搭建机器人”）、“资源管理”“编写程序”“运动回放”功能，其中的几个具体功能如下：

“构建机器人”：搭建机器人，修改已有的机器人结构、参数。

“编写程序”：编写程序和修改已有程序。

“资源管理”：管理用户的场景、机器人、程序和仿真包中“我的作品”。

“构建场景”：构建不同的任务场景。

另外，在线仿真大厅与仿真平台两个窗口可以相互访问，在仿真平台单击按钮，可以返回到仿真大厅；单击按钮直接退出 I RobotQ3D。

### 试试看

1. 在仿真大厅和仿真平台两个窗口之间相互切换和访问，熟悉这两个窗口界面。
2. 在 I RobotQ3D 的两个主窗口中分别单击各功能按钮，打开对应的界面，初步了解每一项功能。






### 3. 体验 IRobotQ3D 的仿真环境

用户注册、登录到 IRobotQ3D 的仿真大厅后, 就可以使用它了。使用工作的流程如图 3.15 所示。



登录 IRobotQ3D, 参考图 3.16 所示的工作流程, 在仿真大厅中查看“新手入门\_循迹踢球”任务要求、算分方法等规则, 在仿真平台中运行“新手入门\_循迹踢球”仿真包, 体验 IRobotQ3D。

第 1 步: 双击桌面上的快捷图标, 输入用户名和密码, 单击 **Go!** 按钮, 登录到仿真大厅。

第 2 步: 单击在线大厅左侧、任务列表中基础试玩前面的“+”图标, 在展开的系列中单击“新手入门\_循迹踢球”任务, 等待任务更新完毕后, 进入房间(参见图 3.13)。

第 3 步: 单击任务导航栏中的 **任务说明**, 工作区显示“新手入门\_循迹踢球”任务描述, 如图 3.16 所示, 拖动工作区右侧的滚动条, 查看此任务要求、算分方法等规则。

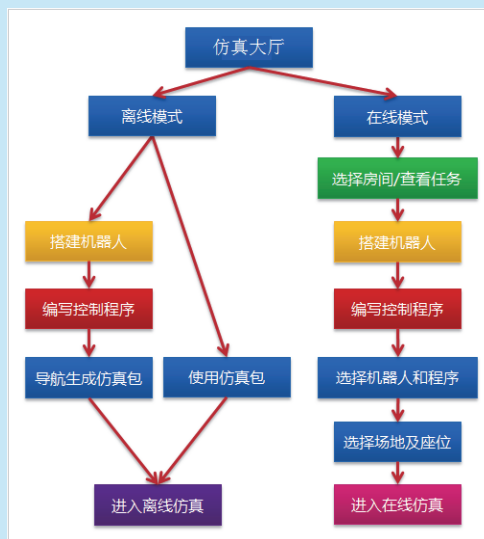


图 3.15 IRobotQ3D 的使用流程



图 3.16 “新手入门\_循迹踢球”任务描述

第 4 步: 在仿真大厅窗口工具栏中单击  按钮, 进入图 3.14 所示的仿真平台, 单击功能区中的“开始仿真”按钮, 打开“快速仿真包”窗口, 如图 3.17 所示。





图 3.17 “快速仿真包”窗口

### 小博士

快速仿真包窗口左侧的仿真包列表窗格中包含萝卜圈“官方作品”仿真包和“我的仿真包”。“官方作品”仿真包中，为用户提供了一组包括任务场景、机器人和控制程序的仿真作品，供我们学习。“我的仿真包”是系统预留的保存个人仿真包的位置。




第 5 步：在任务列表窗格中单击官方作品前面的 ，展开任务列表，选择“05 新手入门\_循迹踢球”仿真包，在任务预览窗格中显示仿真包检测报告，如图 3.18 所示。



图 3.18 “05 新手入门\_循迹踢球”仿真包状态检测报告


### 小博士

仿真包状态检测报告包括“任务”（对应场景）、“机器人”和“程序”，某个项目之后显示“√”，表示对应的文件存在；显示“×”，表示对应的文件不存在，可能被重命名或已删除。

在“仿真包列表”中单击选择一个仿真包，如无提示“仿真包缺少文件”，则此仿真包可用。可查看其状态检测报告，了解此仿真包使用的任务场景、机器人和控制程序，单击“进入仿真”，直接进入仿真。

如提示缺少文件时，单击提示窗口的“确定”，在检测报告中查看缺少的文件。缺少文件的仿真包已不可用，可使用“资源管理”功能将其删除。



第 6 步：单击窗口下方任务导航窗格中的“进入仿真”按钮 ，进入“保存快速仿真包”窗口，单击 **不保存进入**，打开仿真窗口，进入仿真界面，如图 3.19 所示。

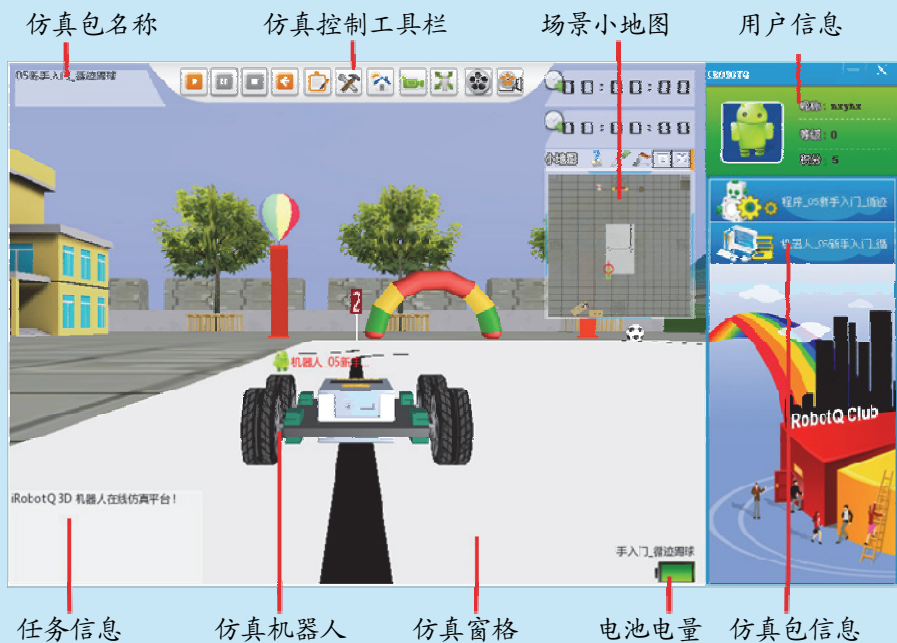


图 3.19 仿真窗口

### 小博士

仿真界面上方有一组工具，用来控制、调试机器人和程序等。将鼠标指针移到相应按钮处停留片刻，就会弹出提示，说明这一按钮的功能。





在仿真开始前，界面上仅显示仿真包名称、机器人信息、当前机器人和控制程序名称、电池电量和场景小地图等，开始仿真后，还会显示仿真时的任务场景名称、机器人部件的参数(如直流电机转速，传感器检测值等)、仿真时间等(参见图 3.20)。




第 7 步：单击仿真控制工具栏中的“开始”按钮，加载程序完成后开始仿真，如图 3.20 所示。



图 3.20 “新手入门\_循迹踢球”仿真截图



第 8 步：仿真完成后，弹出完成任务界面，如图 3.21 所示。单击按钮返回到仿真窗口，单击仿真窗口右上角的按钮退出仿真窗口，返回仿真平台(见图 3.14)。



图 3.21 完成仿真的“恭喜”界面

## 知识窗

表 3.1 仿真工具按钮的功能

图标	名称	作用	说明
	开始	开始仿真	
	暂停	暂停仿真	单击  可继续
	停止	停止仿真	不可继续
	复位	场景、机器人复位到初始状态	单击  可再次开始
	编辑程序	在仿真时编辑控制程序，返回仿真后使用修改过的控制程序	停止或复位后可用
	编辑机器人	在仿真时编辑机器人，返回仿真后使用修改过的机器人	停止或复位后可用
	编辑场景	在仿真时编辑场景，返回仿真后使用修改过的场景元素，但规则脚本的修改在重新进入仿真时才有效	停止或复位后可用，离线时官方作品不可编辑
	拍照	对当前仿真窗口进行拍照，保存在程序安装目录的 DriverTemp 目录中	
	变换视角	在第一人称和第三人称之间切换仿真视角	
	设置方位	在仿真开始之前设置机器人的方位和朝向	未开始或复位后可用

## 试试看

在仿真平台运行官方作品中的其他仿真包，进一步熟悉萝卜圈仿真平台的环境及强大的功能。



## 四、学习实践

IRobotQ3D 是仿真学习机器人的虚拟环境，完成机器人的仿真学习后，我们就可以利用萝卜圈公司提供的教学机器人部件，搭建真实的学习机器人，以后还要使这个机器人有“智慧”，帮助我们完成各项学习任务。这里我们先从熟悉萝卜圈公司提供的教学机器人部件开始。

### 1. 任务

① 表 3.2 是萝卜圈教学机器人的配件清单，对照该清单，查验、辨认各配件。





表 3.2 萝卜圈教学机器人配件清单

编 号	名 称	图 片	功 能
ROBOKID-FR1	主控制器		电脑。负责数据的处理，输出控制指令
QD-DDJ-01	大电机		驱动部件
QD-LZ-01	轮子		行动部件
CG-CP-01	触碰传感器		感知信息
CG-CSB-01	超声波传感器		感知信息
CG-HD-01	灰度传感器		感知信息
CG-GX-01	光线传感器		感知信息
CG-ZA-01	障碍传感器		感知信息
S149a	十字棒 2 倍		连接件
S721	十字棒 3 倍		连接件
S735	十字棒 4 倍		连接件
S736	十字棒 6 倍		连接件
S737	十字棒 8 倍		连接件
S361	双接口		连接件
S633	轴套		连接件
S25	转轴		连接件



编 号	名 称	图 片	功 能
S718	紧转轴		连接件
S360	十字转轴		连接件
S711	单边加长转轴		连接件
S35	圆柱		连接件
S354	斜块		连接件
S62	方杆 2 孔		连接件
S175	方杆 1 十字孔		连接件
S357	方杆 1 孔		连接件
S33	方杆 3 孔		连接件
S639	方杆 5 孔		连接件
S619	方杆 7 孔		连接件
S618	方杆 15 孔		连接件
S654	薄连杆双孔，双十字		连接件
S678	厚连杆 3 孔		连接件
S701	厚连杆 5 孔		连接件
S700	厚连杆 7 孔		连接件
S699	厚连杆 9 孔		连接件
S697	厚连杆 15 孔		连接件
S702	厚连杆 2×4 孔 L 型		连接件





编 号	名 称	图 片	功 能
S703	厚连杆 3×5 孔 L 型		连接件
S705	厚连杆 12 孔，双折		连接件
S64	板 1×2 柱		连接件
S50	板 1×4 柱		连接件
S52	板 1×8 柱		连接件
S475	板 10 柱		连接件
S69	板 2×6 柱		连接件
S17	板 6×8 柱		连接件
S511	板 16×16 柱		连接件
S627	正 4 边形 5 孔		连接件

② 每个小组一台 Robokid-FR1 机器人，对照表 3.2，观察组成机器人的各个部件，并在表 3.3 中记录相关信息。

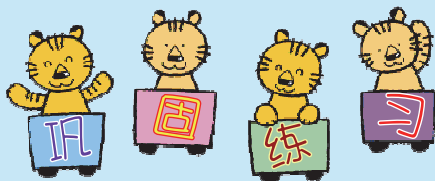
表 3.3 查验机器人记录单

机器人的名称:

部件名称	数量	作用	备注

## 2. 组织实施

6~8 个同学组成 1 个小组，在机器人实验室操作。



1. 判断下列说法的正误。

① 机器人是一种能自动工作的机器装置，一般由执行机构、驱动装置、检测装置、控制系统和复杂机械等组成。( )

② 机器人的工作原理就是利用各种传感器从环境获取信息，并将这些信息传送到机器人的控制系统，控制器对接收到的信息进行判断，做出决定，然后控制机器人动作。( )

③ 所谓机器人就是其外貌和功能都与人类一样。( )

④ IRobotQ3D 有两个主窗口——仿真大厅和仿真平台，其中仿真大厅是个离线平台，适用于在单机模式下进行仿真测试；仿真平台是在线平台，适用于在线仿真测试与竞赛。( )

⑤ 不再使用的仿真包、场景、机器人和程序都可以使用“资源管理”功能删除。( )

2. 下列各项中，属于工业机器人的是( )。

- |           |         |
|-----------|---------|
| A、自动售票机   | B、智能红绿灯 |
| C、数控切割机械手 | D、服务机器人 |

3. 下列各项中，属于特种机器人的是( )。

- |         |         |
|---------|---------|
| A、喷涂机器人 | B、点焊机器人 |
| C、搬运机器人 | D、娱乐机器人 |

4. 下列各项中，机器人系统不包括的是( )。

- |        |        |
|--------|--------|
| A、感知系统 | B、控制系统 |
| C、安全系统 | D、执行系统 |

5. 登录 IRobotQ3D，在仿真大厅上单击“搭建机器人”或在仿真平台上单击“构建机器人”，打开“选择智能设备”界面，选择后单击，打开 Robokid 机器人编辑窗口，在左侧窗格单击“模型”标签卡，查看各模块的部件，了解各部件的主要功能和作用。

6. 请把你想象或设计的机器人画下来，并说明它的用途。





## 阅读材料

## 机器人的发展

1920 年捷克斯洛伐克剧作家卡雷尔·查培克(Karel Čapek, 1890~1938)创作了寓言剧剧本《罗莎姆万能机器人公司》，书中有这样的情节：“未来的某时，某岛屿上的罗莎姆公司大量地制造出了外貌像人的‘人造人’——Robota。从秘书到劳役，它们都能与人共事，为人服务。”在这里，“人造人”第一次有了自己的名字：“Robota”。援引这个捷克语的读音，英、法、意文均写作：“Robot”，我国将它意译为“机器人”。

机器人的历史可以追溯到 20 世纪以前。

18 世纪，法国人雅克·德·沃坎逊设计出世界上最古老的机械机器人——由打孔机控制的织布机。随后，贾卡在其基础上进行了改进，使其结构变得更复杂并减少了人工干预过程，得到如图 3.22 所示的织布机。

1834 年，查尔斯·巴贝奇把“打孔定位”这一设想引入他设计的分析机，这是世界上最早的通用机械数字计算机，如图 3.23 所示，它为电子计算机的发明奠定了基础，巴贝奇也因此被称为“计算机之父”。

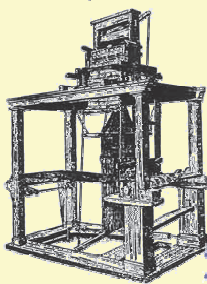


图 3.22 贾卡织布机

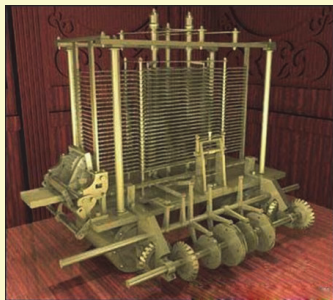


图 3.23 后人复制的巴贝奇分析机

1954 年，美国人乔治·沃尔制造出世界上第一台电子可编程控制的机器人。

1992 年，日本的 150 位工程师历时 11 年，耗资 8000 万美元研制出一台微型机器人，它的体积不到 3 立方厘米，质量为 1.5 克。这个机器人有三维视觉，头部能自如转动，能躲开障碍物，能改变方向，被推撞后可以自我平衡，可以照料人和完成多种危险艰苦的工作。

1997 年，日本本田公司研制出世界上第一台可以像人一样行走的高 1.6 米的机器人，这是机器人发展史上的一个里程碑。

机器人技术作为 20 世纪人类最伟大的发明之一,自上世纪 60 年代初问世以来,经历几十年的发展已取得长足的进步,现在的机器人可谓千姿百态,有的“有手无头”,有的“三头六臂”,个个“心灵手巧”,应用涉及到各个领域,真可谓“可上九天揽月,可下五洋捉鳖”。

2004 年 1 月,美国发射的“勇气”号(见图 3.24)和“机遇”号火星车机器人先后在火星成功登陆。火星车在火星表面行走、拍摄、钻探、化验,非常精彩地完成了自己的使命。

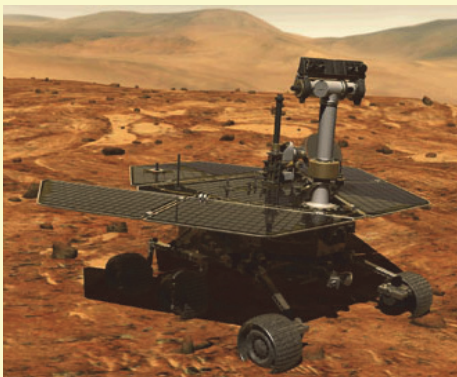


图 3.24 美国“勇气”号火星车

2013 年 12 月,中国首辆月球车“玉兔号”(见图 3.25)顺利驶抵月球表面。玉兔号月球车设计质量 140 千克,能源为太阳能,能够耐受月球表面真空、强辐射、摄氏零下 180 度到零上 150 度极限温度等极端环境。月球车具备 20 度爬坡、20 厘米越障能力,并配备有全景相机、红外成像光谱仪、测月雷达、粒子激发 X 射线谱仪等科学探测仪器。

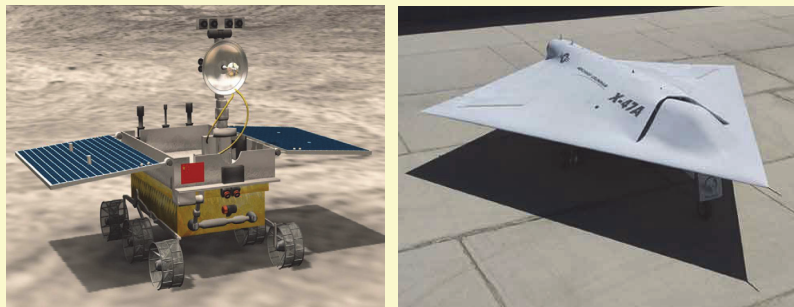


图 3.25 中国玉兔号月球车和新型无人机

参考资料网址: <http://www.bdxx.net/kx/kexue-2/zutiweb/zu40/01/000.htm>





## 第4单元 搭建机器人

学习机器人，不论是在仿真环境还是在真实环境中，每一个任务的完成都离不开机器人。因此，我们首先要学会搭建机器人。

IRobotQ3D 仿真平台提供了机器人搭建平台。在仿真大厅或仿真平台中，单击“搭建机器人”或“构建机器人”图标，打开“选择智能设备”界面，有两种机器人搭建环境可供选择，如图 4.1 所示。




在“选择智能设备”界面中，选择其中一个环境后，单击就能打开对应的机器人编辑窗口：选择，支持 IROBOTQ 虚拟机器人搭建；选择，支持与 Robokid FR1 实物机器人匹配的虚拟机器人搭建。由于本书采用虚实结合环境学习机器人，因此在这里选择



图 4.1 “选择智能设备”界面

，打开 Robokid 机器人编辑窗口，如图 4.2 所示。

Robokid 机器人编辑窗口除 Windows 应用程序常有的工具栏和窗口控制按钮外，主要由“模型库”“属性面板”“机器人编辑区”“视角面板”四个区域组成，每个区域的功能如下：

① **模型库**：由“模型”“工作台”和“模板”三个标签卡对应的三个面板组成，其中：

模型——包含控制器、驱动、安装块、传感器、其它 5 个大类的所有模型。

工作台——显示所有添加到编辑区中的模型及它们之间的关系。

模板——显示用户创建的模板，用模板可保存经常使用的模型组合。

② **属性面板**：由“属性”“备注”和“机器信息”三个标签卡对应的面板组成，其中：



属性——设置直流电机、伺服电机、传感器的属性。

备注——编辑和查看机器人的用户备注信息。

机器信息——查看机器人零部件的数量，机器人的重量、尺寸等信息。

③ **机器人编辑区**：构建机器人，完成机器人零部件的安装等操作。

④ **视角面板**：调整查看编辑区的视角，完成视角的环绕、缩放、平移等操作。

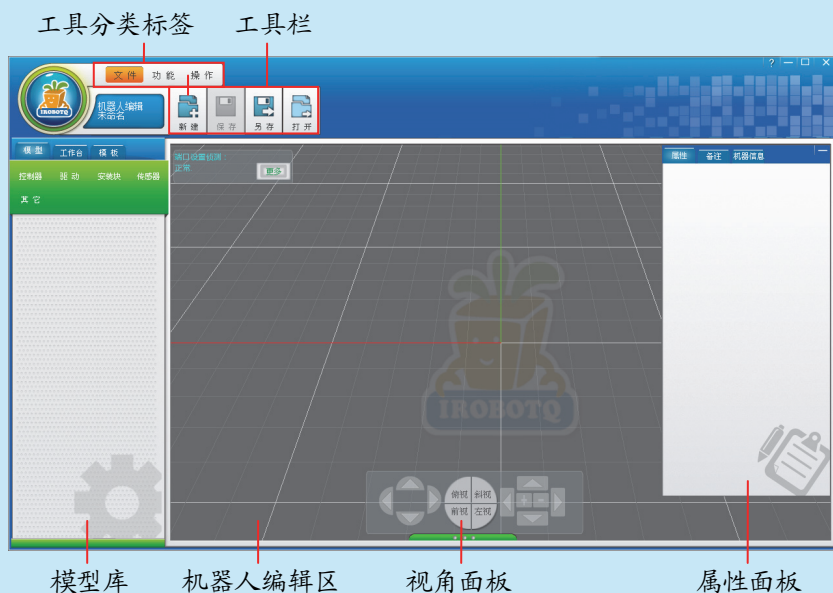


图 4.2 Robokid 机器人编辑窗口

## 一、构建机器人部件

机器人是由一个个零部件组装而成的，不论是仿真环境还是真实实验，IRobotQ3D 给我们提供的都是一些最基本的基础零部件。因此，在具体任务中，需要根据任务需求，先将基础零部件构建(组装)成需要的机器人部件。在 IRobotQ3D 的机器人编辑窗口，可以完成机器人部件的构建，完成后保存为“模板”，在搭建机器人时可以直接使用。

在这册书的学习中，我们要使用的轮式车形机器人的主体，由作为车身的控制器、驱动装置、车轮和辅助轮几个主要部件组成，我们先完成这些部件的构建。




表 4.1 是 IRobotQ3D 轮式车形机器人驱动装置的零部件清单，尝试在 Robokid 机器人编辑窗口完成“左驱动装置”的构建，以“左电机模板”备注添加到模板中。






表 4.1 轮式车形机器人左驱动装置的零部件清单

名 称	图 片	数量	作 用
大电机		1	驱动
厚连杆 7 孔		1	提供控制器与其他零部件的连接
厚连杆 3×5 孔 L 型		1	提供电机与控制器、控制器与其他部件的连接孔
十字棒 3 倍		1	提供电机与轮子的连接
转轴		9	用于电机与厚连杆 7 孔、厚连杆 7 孔与厚连杆 3×5 孔 L 型、厚连杆 3×5 孔 L 型与控制器、厚连杆 5 孔与正交连轴器双孔单十字以及厚连杆 5 孔与导向轮的连接
正交连轴器双孔单十字		1	组合后作为导向轮左侧安装支架与电机连接
厚连杆 5 孔		1	
十字转轴		2	
双接口		1	

第 1 步：登录 IRobotQ3D，在仿真大厅工具栏中单击  按钮进入仿真平台，然后单击  打开“选择智能设备”界面，选择  后单击  打开 Robokid 机器人编辑窗口。

### 金钥匙

也可以在仿真大厅工具栏中，单击  按钮进入机器人编辑窗口，但这样进入后程序工作在在线模式，需要利用网络传输数据，工作效率受网络速度的影响。从仿真平台中进入后，工作在离线模式，程序直接在本机上运行，不受网络速度影响，工作效率会高一些。



第 2 步：单击模型库的“模型”面板中的 **驱动**，在“驱动”模型列表中选择“大电机”，将鼠标指针移到编辑区绿色与红色坐标线相交处后单击，把大电机放到编辑区中；单击“模型”面板中的 **安装块**，在“安装块”模型列表中选择“十字棒 3 倍”，放置到编辑区“大电机”左侧，如图 4.3 所示。

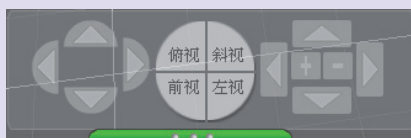




图 4.3 添加“大步进电机”和“十字棒 3 倍”部件

### 金钥匙

1. 在 IRobotQ3D 模型库的模型列表中，将鼠标指针移到某一模型上，系统会给出该模型的名称。
2. 属性面板和视角面板都是浮动面板，如果影响对机器人的编辑，可以分别将它们最小化，其中视角面板的最小化/还原按钮是有三个白点的绿色区域，如图 4.4 所示。



最小化/还原按钮  
图 4.4 视角面板



第 3 步：在编辑区单击“十字棒 3 倍”右端的安装点，橘色安装点变为粉色，接着单击“大电机”输出轴中间的黄色安装点，“十字棒 3 倍”被安装到“大电机”输出轴上，如图 4.5 所示。

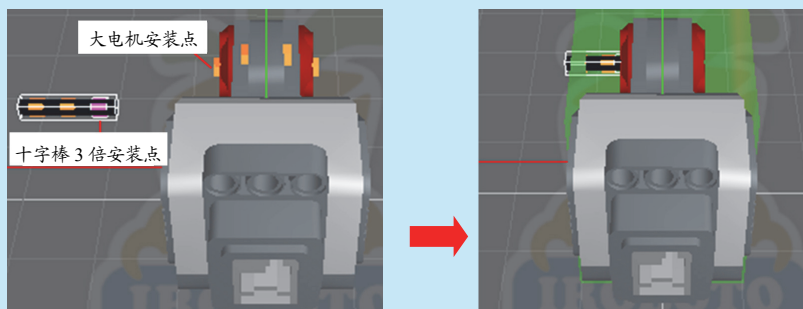


图 4.5 安装“十字棒 3 倍”到“大电机”上

### 金钥匙

如果安装点选择有误，可以按键盘上的 **Esc** 键取消。





第4步：在“安装块”模型列表中，分别选择2次“转轴”和1次“厚连杆7孔”，依次添加到编辑区“大电机”右侧；在编辑区单击“大电机”，按下鼠标右键向左拖动到显示没有“十字棒3倍”的一侧；然后参照第3步和图4.6，分别将2个“转轴”依次安装到“大电机”顶端两个安装点中；最后参照第3步和图4.7所示，将“厚连杆7孔”右前方安装到“大电机”顶端左数第一个“转轴”上。

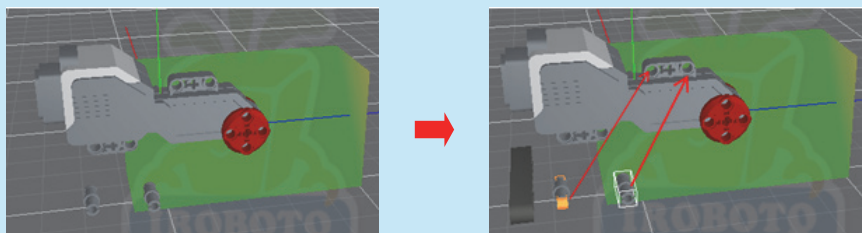


图 4.6 添加2个“转轴”和1个“厚连杆7孔”及2个“转轴”安装位置

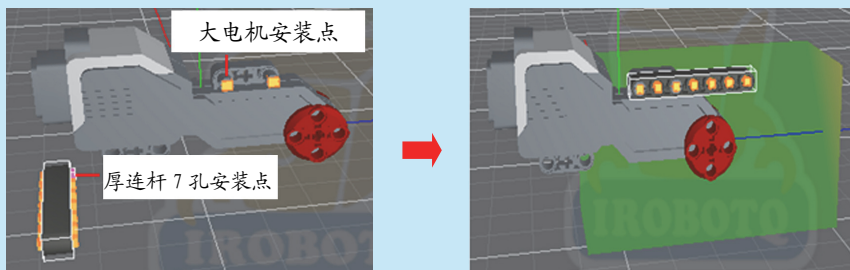


图 4.7 安装“厚连杆7孔”到“大电机”上

### 金钥匙

在机器人编辑窗口可以快速改变模型的视角，操作方法是：选中模型后按下鼠标右键，左右拖动可以切换水平方向的视角，上下拖动可以切换垂直方向的视角。



第5步：在“安装块”模型列表中，分别选择4次“转轴”和1次“厚连杆3×5孔L型”依次添加到编辑区：参照图4.8，将其中2个“转轴”分别安装到“厚连杆7孔”右数第1和第3个安装孔中，再将“厚连杆3×5孔L型”拐角处安装孔与“厚连杆7孔”右数第2个转轴连接，接着按3次空格键，使“厚连杆3×5孔L型”长杆转270度后安装到“厚连杆7孔”上；最后参照图4.9所示，将剩下的2个“转轴”分别安装到“厚连杆3×5孔L型”上数第1和第3个安装孔中。

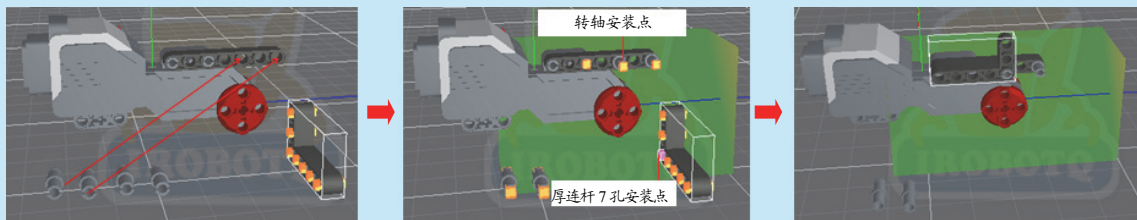


图 4.8 安装“厚连杆3×5孔L型”到“厚连杆7孔”上



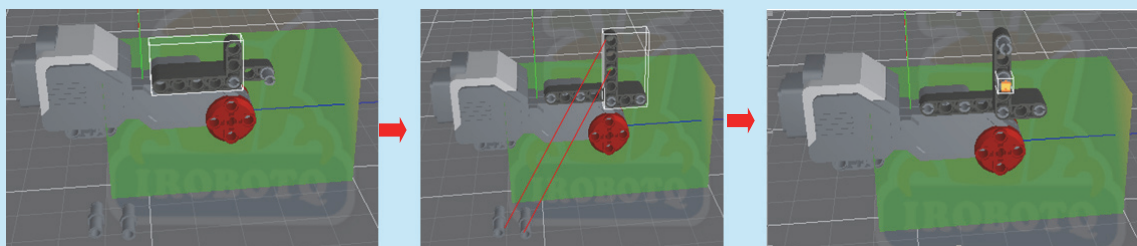


图 4.9 安装 2 个“转轴”到“厚连杆 3×5 孔 L 型”上

### 金钥匙

在 IRobotQ3D 机器人编辑界面，选中模型后按空格键可以改变模型方向，系统默认：按 1 次空格键，模型在水平方向逆时针旋转 90 度，按 4 次后转过一周，按第 5 次在垂直方向旋转，显示模型底部，按第 6 次回到初始位置（显示模型顶部）。但针对柱形部件按 4 次空格键旋转一周后，由于柱形部件相对于初始位置它的曲面没有上下区别，再按第 5 次和第 6 次空格键时，模型视角不发生变化。



第 6 步：在“安装块”模型列表中，分别选择 2 次“十字转轴”、3 次“转轴”以及“双接口”、“正交连轴器双孔单十字”、“厚连杆 5 孔”各 1 次，依次添加到编辑区，如图 4.10 所示。

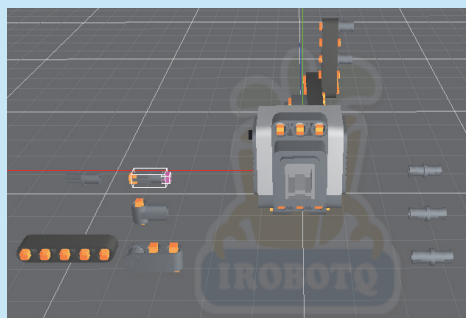
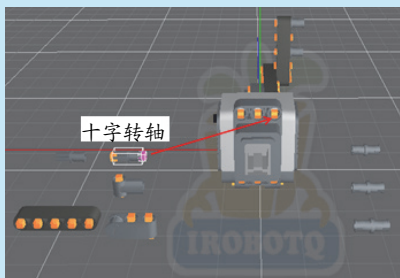


图 4.10 添加多个模型

第 7 步：在编辑区单击“大电机”，按一次空格键，使“大电机”转过 90 度显示尾部三个安装孔，然后参照图 4.11，依次将一个“十字转轴”圆形安装点安装到“大电机”尾部右数第 1 个安装孔中，将“双接口”安装到“十字转轴”的十字安装点上，将另一个“十字转轴”圆形安装点连接到“双接口”上方安装点上，将“正交连轴器双孔单十字”的十字安装孔安装到“十字转轴”的十字安装点上，最后按 2 次空格键，使“正交连轴器双孔单十字”转 180 度后安装到“大电机”外侧。



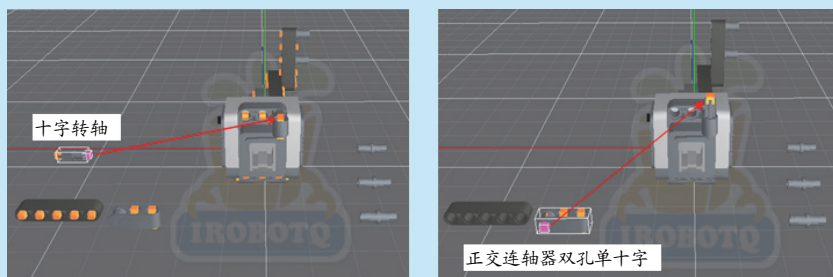


图 4.11 安装示意图

**金钥匙**

1. 可以通过上下拨动鼠标滚轮，放大或缩小编辑区上模型的大小，方便观察各个模型的安装点。
2. 如果编辑区中某个模型距离安装位置太远或太近，可以先单击选中它，再按下鼠标左键拖动至合适位置。



第 8 步：在编辑区单击“大电机”，按下鼠标右键向左拖动，显示“正交连轴器双孔单十字”右侧安装点，参照图 4.12 将一个“转轴”安装到“十字转轴”外侧第 1 个安装孔中，再将“厚连杆 5 孔”与“转轴”连接，然后按 1 次空格键，让“厚连杆 5 孔”转旋 90 度后安装在“正交连轴器双孔单十字”上。

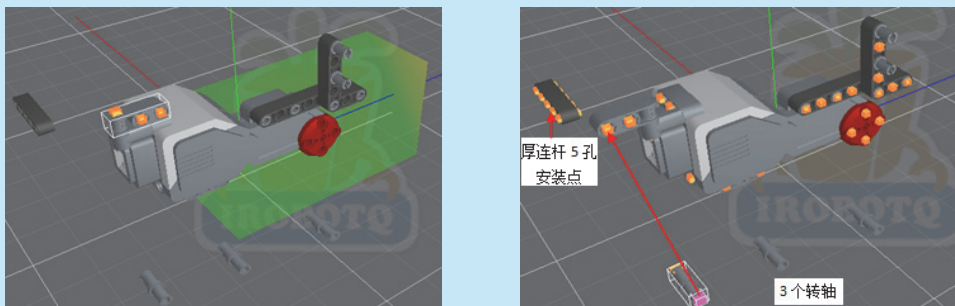



图 4.12 “转轴”安装示意图

第 9 步：参照第 8 步操作，分别将剩余的两个“转轴”依次安装到“厚连杆 5 孔”上数第 1 和第 3 个安装孔中，如图 4.13 所示。

第 10 步：在编辑区单击“大电机”，然后单击模型库的“模板”标签，在打开的“模板”面板中单击“创建模板”按钮，组合后的部

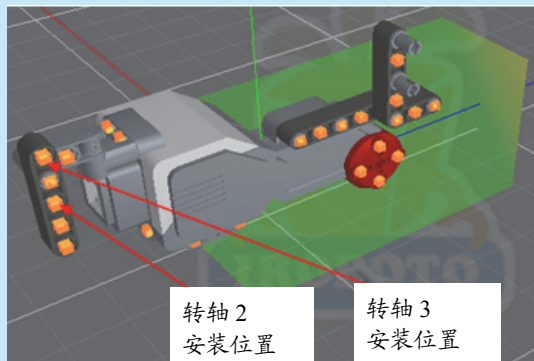



图 4.13 “转轴 2”和“转轴 3”安装位置示意图



件被添加到模板列表中，如图 4.14 所示；最后单击“模板”面板中的“添加备注”按钮，在弹出的列表框中输入“左电机模板”，如图 4.15 所示。

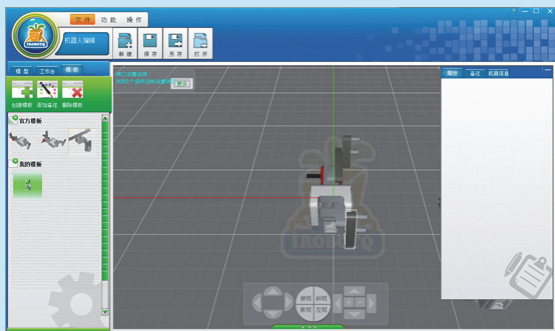


图 4.14 创建左电机模板

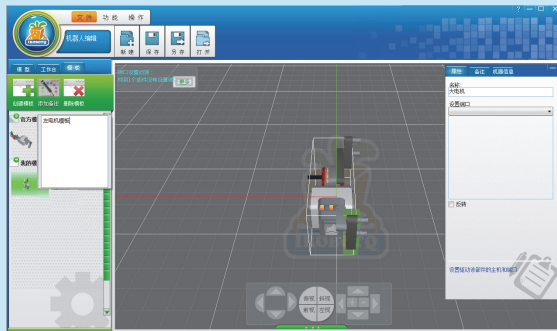



图 4.15 添加备注“左电机模板”

### 小博士

1. 在左侧的模板列表窗格中，包括“官方模板”和“我的模板”两个文件夹，“官方模板”是系统提供的一组搭建机器人时常用的模板，“我的模板”是专门为个人创设模板预留的文件夹。
2. 创建的模板可以多次使用，以简化重复性部件构建的过程。



第 11 步：单击 RoboDK 机器人编辑窗口右上角的，返回仿真大厅。

### 知识窗

#### 视角面板

萝卜圈仿真是 3D 形式的，在机器人编辑平台上可以从不同的方向观察机器人模型。利用机器人编辑窗口的“视角面板”（如图 4.16 所示）可以改变编辑窗口中模型的视角，具体功能如下。

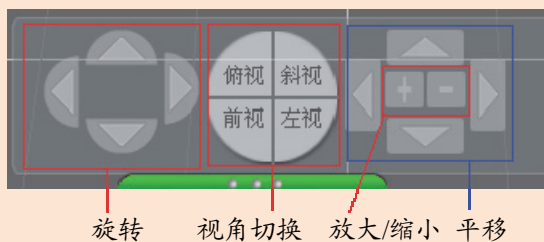


图 4.16 视角面板





视角切换：可选择“俯视”“斜视”“前视”和“左视”四种方向观看。

环绕：可以模型为中心，环绕模型查看。

缩放：对模型放大(+)和缩小(-)。

平移：可以沿上、下、左、右方向平移模型。



### 说说看

我们学习了几种调整安装视角的方法？它们都分别适用于什么情况？



### 试试看

同桌分工合作，分别负责为机器人构建一个右驱动装置和辅助轮装置，并以“右电机模板”和“辅助轮模板”为备注，添加到 Robokid 机器人编辑窗口的模板中，然后共享。



## 二、搭建仿真机器人

完成机器人部件的构建后，再搭建机器人就方便多了，只要在 IRobotQ3D 的 Robokid 机器人编辑窗口中，将需要的部件组装起来就可以了。



### 做一做

搭建一个两轮驱动的轮式车形机器人作为基础仿真机器人，并以“学习者号”为名保存。

第 1 步：登录 IRobotQ3D，在仿真大厅打开 Robokid 机器人编辑窗口。

第 2 步：单击模型库的“模型”面板中的 **控制器**，打开“控制器”面板，在列表窗格中单击“控制器”，将鼠标指针移到编辑区绿色与红色坐标线相交处后单击，把控制器放到编辑区中，如图 4.17 所示。

第 3 步：单击模型库的“模板”标签，在“模板”列表中单击 **我的模板** 中的左电机模板，将鼠标指针移到编辑区控制器左侧单击，添加左电机部件；用同样的方法添加右电机部件和辅助轮部件到编辑区，如图 4.18 所示。

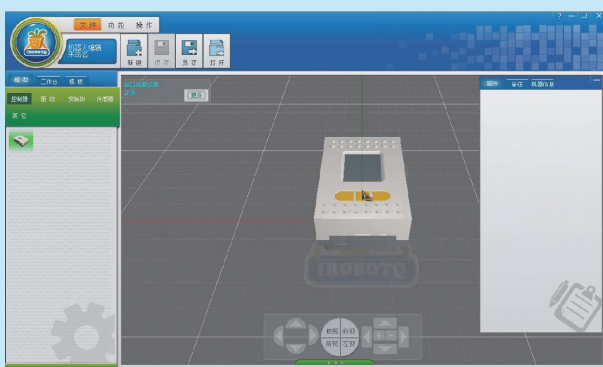


图 4.17 添加机器人控制器

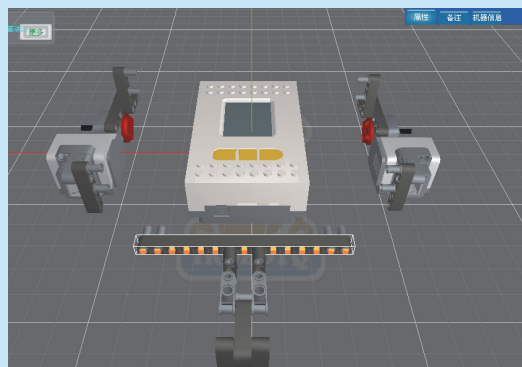


图 4.18 添加左、右电机部件及辅助轮部件

### 金钥匙

通过“模板”列表添加左、右电机部件及辅助轮部件时，也可以通过“官方模板”进行添加，因为系统在“官方模板”中也提供了相应的模板。



第 4 步：在编辑区单击右电机部件上的 5 孔厚连杆左上方的转轴，使其出现橘色安装点，再次单击使其变为粉色；然后单击选中控制器，按下鼠标右键缓缓向左拖动，显示其右侧安装点，再单击控制器右上角的安装点，右电机部件被安装到控制器上，最后按 1 次空格键，使右电机部件旋转 90 度后安装在控制器上，如图 4.19 所示。

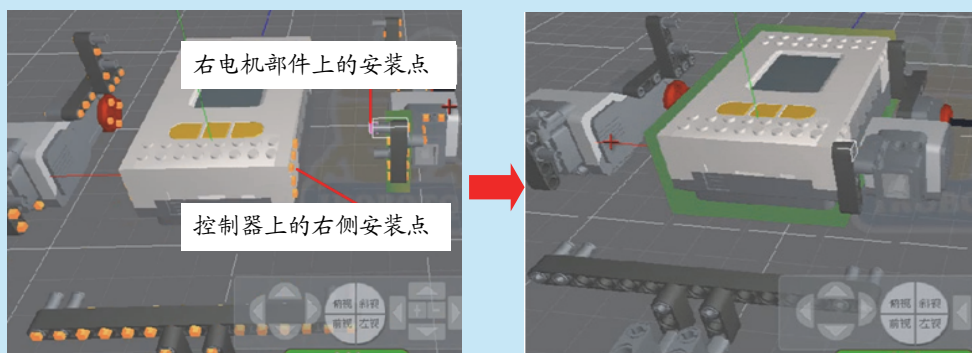


图 4.19 将右电机部件安装到控制器上

第 5 步：参照第 4 步，将左电机部件安装到控制器上，如图 4.20 所示。

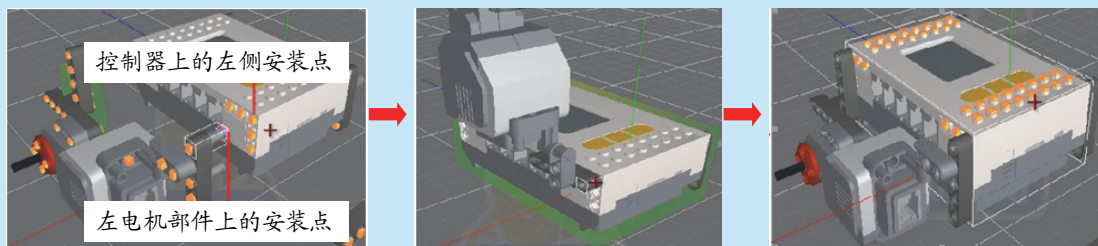


图 4.20 将左电机部件安装到控制器上





### 小博士

由于编辑机器人用到的工具种类较多，所以机器人编辑窗口工具栏采用了“工具分类标签”和“工具栏”结合的管理办法，选择不同的标签，工具栏上会打开一组相应的工具，如图 4.21 所示。



图 4.21 不同标签对应的工具栏

第 6 步：在编辑区单击辅助轮部件，显示橘色安装点，再单击左侧安装点，使其变为粉色；然后单击控制器，按下右键缓缓向上拖动，显示左电机部件有三个安装点的尾部；最后单击左电机模板尾部中间安装点，将辅助轮部件安装到电机上，如图 4.22 所示。

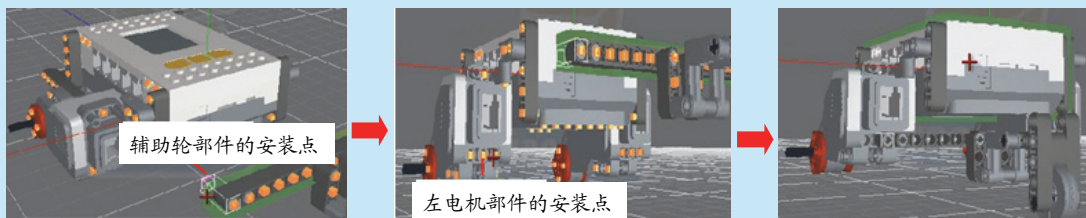


图 4.22 将辅助轮部件安装到电机上

### 小博士

安装辅助轮的目的在于平衡方向和防止左、右电机从控制器上脱落。



第 7 步：在模型库“模型”面板中单击 **驱动**，在列表中 2 次选择轮子添加到编辑区；参照图 4.23，在编辑区旋转控制器的安装视角，分别在左、右电机十字棒 3 倍上安装轮子。



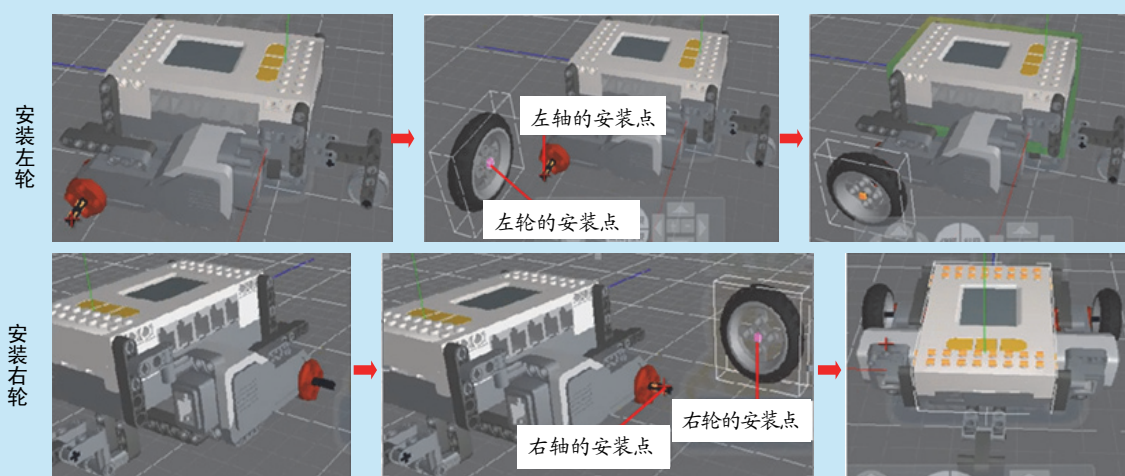


图 4.23 安装左、右轮

第 8 步：在编辑区分别选择控制器上的左电机和右电机，单击属性面板中的“属性”标签，进入属性面板，分别设置它们的属性(如图 4.24 所示)：

左侧电机：“名称”设为“大电机\_左”，“设置端口”设为“1”；

右侧电机：“名称”设为“大电机\_右”，“设置端口”设为“2”。

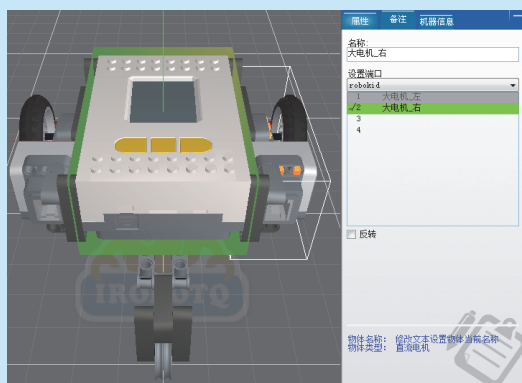



图 4.24 设置左、右电机的端口

### 小博士

为了让控制器能更明确地将指令下达到各个设备，必须给每个与控制器连接的通电设备(如电机、传感器等)分配唯一的端口，通常大电机的可用端口范围是 1~4。

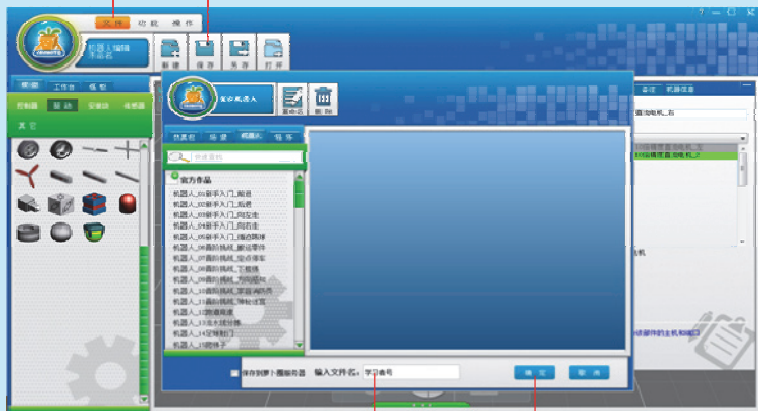


第 9 步：单击“文件”标签，在打开的工具栏中单击“保存”按钮 ，打开“保存机器人”对话框，输入文件名“学习者号”，单击 **确定** 按钮，保存机器人，如图 4.25 所示。





1.单击“文件” 2.单击“保存”



3.输入文件名 4.单击“确定”

图 4.25 保存机器人

### 说说看

为什么不给“学习者号”控制器上的辅助轮设置端口？



### 试试看

将“学习者号”机器人创建为模板。



## 三、学习实践

IRobotQ3D 软件提供了仿真的机器人学习平台 IRobotQ3D，还有与之配套的机器人学习器材(第3单元“学习实践”任务中已经查验)，利用这套学习器材，我们可以根据任务需求，搭建相应的学习机器人。

机器人是任务的活动主体，它的结构对完成任务影响非常大。因此，搭建机器人需要经历规划设计、准备材料、组装调试等环节，具体工作如下。

① 规划设计：确定机器人要实现什么功能，设计机器人通过哪些动作来实现相应的功能。

② 学习装配说明书。

③ 准备材料：按照机器人的动作要求和执行环境准备材料。此时应该对材料的组合方式、动作的实现原理有清晰的构思。

④ 组装机器人：用材料组装出机器人实体，记录组装过程。

提示：可以先组装机器人的常规部件，再将各部件与控制器组装成机器人。



⑤ 调试：对机器人进行调试、修改、完善。

### 1. 任务

使用“萝卜圈”学习机器人套件，参照本单元中的“做一做”，自己动手搭建一个学习机器人，命名为“学习者号”。

### 2. 组织实施

① 6~8 个同学组成 1 个小组，在机器人实验室操作。

② 操作步骤：

第 1 步：规划设计，明确用途，设计结构。

在第 3 单元的“学习实践”活动中，我们观察了已组装好的 Robokid-FR1 实物机器人，大致了解其结构和所用部件，这里我们先组装一个常见的两轮机器人。

第 2 步：准备材料，参照表 4.2 列出材料清单（表格行不够可自行添加）。

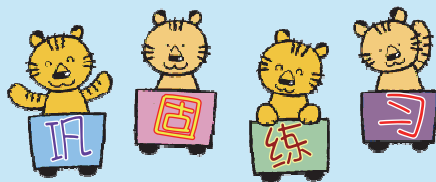
表 4.2 两轮机器人材料清单

编号	名 称	作 用

第 3 步：组装机人的左、右电机部件和辅助轮部件。

第 4 步：将左、右电机部件和辅助轮部件安装到电机上，然后安装上车轮。

第 5 步：调试。



1. 判断下列说法的正误。

① 在 Robokid 机器人编辑窗口中搭建机器人时，产生粉色安装点的部件一定是主动部件，移动后会被“镶嵌”到指定安装点上。（ ）

② 虽然机器人是由一个个基础零部件组装而成的，但是在搭建机器人时，我们总是根据具体任务，先将基础零部件构建成一个独立的机器人部件，再对这些部件进行组装，搭建出最终的机器人。（ ）

③ IRobotQ3D 仿真平台提供了两种虚拟机器人的搭建平台。（ ）

④ IRobotQ3D 仿真平台在离线模式下工作，程序会直接在本机上运行，不会受网络速





度的影响,工作效率会提高。( )

⑤ 在 IRobotQ3D 仿真平台搭建机器人时,需要给所有部件设置端口。( )

2. 下列 IRobotQ3D 轮式车形机器人驱动装置的零部件中,不起连接作用的是( )。

A、大电机

B、转轴

C、十字棒

D、厚连杆

3. 通过仿真大厅进入 Robokid 机器人编辑窗口和从仿真平台进入 Robokid 机器人编辑窗口,工作效率( )。

A、一样

B、高一些

C、高或低要看具体情况

D、稍低一些

4. 在 Robokid 机器人编辑窗口,下述选项中不能对已选择的模型改变方向或视角的是( )。

A、按键盘上的空格键

B、按下鼠标左键,上、下、左、右拖动

C、按下鼠标右键,上、下、左、右拖动

D、利用视角面板

5. 在 IRobotQ3D 模型库的模型列表中,依次打开控制器、驱动、安装块、传感器、其它 5 大类型模型列表,将鼠标指针分别移到模型列表的各模型上,观察和了解每一个模型的名称及功能,初步了解各个模型的归属类别。

6. 说一说,在 IRobotQ3D 的 Robokid 机器人编辑窗口中为什么要创建模板?交流一下在 IRobotQ3D 软件中搭建机器人的经验。

7. 试着在 IRobotQ3D 机器人编辑窗口中搭建你心中的机器人(图 4.26 可供参考),说出你希望你的机器人能胜任哪些任务。



图 4.26 形形色色的机器人

8. 比较一下在 IRobotQ3D 软件中搭建机器人和在实验室搭建实物学习机器人,说一说有什么异同?

## 第5单元 安全出站

我们已经完成“学习者号”机器人的仿真搭建和实物组装，它是不是可以执行任务了？图 5.1 是 IRobotQ3D 提供的名称为“安全出站”的仿真场地，场景是一个公交车站，要求汽车机器人从起始位置沿箭头所示方向驶出，先到达蓝色色块所示检修台上进行检修，完成检修后驶出红色出站口。



图 5.1 “安全出站”仿真场地

显然“学习者号”还不能自主完成这个任务，目前它只是一个没有“智慧”的机器，需要我们“赋予”它智慧。

机器人的大脑——“控制器”是一个微型计算机(简称单片机)，懂得由 0 和 1 两个符号组成的机器语言指令，因此，我们可以使用计算机语言编写程序，然后让电脑为我们当翻译，把我们的“指令”传达给机器人，机器人接收了这些指令后就会按照程序中的指令去完成任务。编写程序并通过电脑把它传输给机器人，就是赋予了机器人智慧，如图 5.2 所示。

IRobotQ3D 中的“编写程序”窗口，就是系统提供的为机器人编写程序的平台，利用它可以方便地与机器人进行交流。

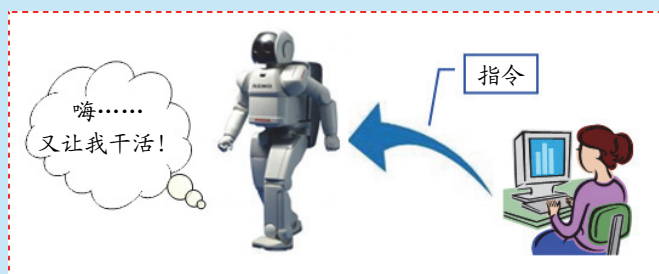


图 5.2 指挥机器人工作

## 一、熟悉 IRobotQ3D 的编程环境


在 IRobotQ3D 的仿真大厅或仿真平台中,单击“编写程序”功能图标,打开“选择编程平台”界面,如图 5.3 所示。选择编程语言后单击,就能打开程序编写窗口。这里选择 ROBO-LOGO,打开“LOGO 程序编写”窗口,如图 5.4 所示。



图 5.3 “选择编程平台”界面



图 5.4 LOGO 程序编写窗口

LOGO 程序编写窗口,除了有与机器人编辑窗口类似的工具栏和窗口控制按钮外,主要由三个区域和一个浮动面板组成,各区域的功能如下。

常用函数窗格:常用的 LOGO 函数、流程及命令列表。

程序编辑区：编写、修改 LOGO 程序的区域。

工具箱窗格：含自定义函数列表窗格和快速文本复制粘贴窗格。在自定义函数列表窗格中，显示所有用户定义的函数(过程)；快速文本复制粘贴窗格用于临时保存多条不同命令或文本，以方便多次粘贴，类似于剪贴板。

浮动面板：主要用于在程序编辑区查找命令或文本。

由于 IRobotQ3D 是一个仿真环境，在它的 LOGO 程序编写窗口编写 LOGO 程序，与第 1、2 单元在 LOGO 语言编程环境下编写 LOGO 程序(即过程)有所不同，因为 LOGO 语言编程环境是不需要考虑外部环境因素影响的理想编程环境，而在 IRobotQ3D 仿真环境中，需要根据真实环境中机器人部件执行指令的实际情况来修正指令，例如，作为机器人驱动部件的电机在执行向前运动指令后，由于惯性的原因，不可能立即再执行向后运动的指令，需要先让电机停下来后，再执行其他运动类指令。

这里以本单元开始给出的任务为例，对用自然语言表示的算法设计、LOGO 语言编程和仿真平台编程进行比较，结果如表 5.1。

表 5.1 自然语言、LOGO 语言、仿真平台对同一案例的算法设计和程序设计

步骤	用自然语言表示的算法设计	LOGO 语言编程	仿真平台编程
1	从起始位置向前驶入直角弯道	FD 850	FD 850 WAIT 500
2	右转 90 度	RT 90	RT 87
3	从直角弯道向前到检修台(蓝色色块)位置	FD 1000	FD 1000 WAIT 500
4	从检修台(蓝色色块)后退至直角弯道	BK 1000	BK 1000
5	向左转 90 度	LT 90	LT 87 WAIT 500
6	从直角弯道驶出红色出站口	FD 1700	FD 1700

从表 5.1 可以看出，同一个动作，在 LOGO 语言中编写的程序和仿真平台编写的程序，虽然所用命令及语法基本一致，但在仿真环境中，需要根据机器人在实际环境运行受到的外在因素影响，而进行适当的修正。

例如，对表 5.1 中第 1 步“从起始位置向前驶入直角弯道”的编程，考虑机器人自身的惯性，需要给出 0.5 秒的等待时间，使机器人电动机改变运动方向，为此要用到 LOGO 语言的“等待”命令“WAIT”，它的具体内容如表 5.2 所示。

表 5.2 “等待”命令

命 令	格 式	作 用
等待(WAIT)	WAIT n	等待 n 毫秒

又例如，对表 5.1 中第 2 步“右转 90 度”的编程，在仿真环境中“RT”命令的参数值不一定是 90，而要根据实际的路况、机器人的重量及驱动器性能等因素，进







行相应的修改。

综上所述,在仿真平台中编写 LOGO 程序,应该遵守 LOGO 语言程序设计的一般原则,并结合真实环境中机器人运行的实际情况,编程的一般步骤为:



## 二、体验 IRobotQ3D 仿真环境编程

为了方便我们学习,IRobotQ3D 提供了一组汽车运动场景,图 5.1 就是其中一例。从本节开始,我们将以一个驾驶员的身份,根据任务的需要编写程序,体验给我们的学习机器人——“学习者号”赋予“生命”与“智慧”,完成具体任务的全过程。



参照图 5.1 描绘的场景和本单元开始给出的任务描述,在 LOGO 程序编写窗口中编写程序,指挥“学习者号”机器人在 IRobotQ3D 仿真平台完成测试。程序调试完成后,以“安全出站”为名保存为仿真包。





### 1. 任务分析


“学习者号”从起始位置沿箭头所示,先向前驶入直角弯道,右转 90 度,再向前到达检修台(蓝色色块位置),然后后退到直角弯道处,左转 90 度,最后沿箭头向前驶出红色出站口。



### 2. 算法描述

- ① 从起始位置向前驶出 850 步至直角弯道。
- ② 等待 0.5 秒(为改变运动方向做准备),向右转 90 度。
- ③ 从直角弯道向前驶出 1000 步至检修台(蓝色色块位置)。
- ④ 等待 0.5 秒(为改变运动方向做准备),从检修台后退 1000 步至直角弯道。
- ⑤ 向左转 90 度,等待 0.5 秒(为改变运动方向做准备)。
- ⑥ 从直角弯道驶出红色出站口。

### 3. 编写程序与仿真调试

第 1 步:登录 IRobotQ3D,在仿真大厅工具栏中单击  按钮进入仿真平台,然后单击  打开“选择编程平台”界面,选择  后单击 ,打开 LOGO 程序编辑窗口。

第 2 步:单击“文件”标签,在打开的工具栏中单击 ,LOGO 程序编辑窗口被清空,单击程序编辑区,输入程序,如图 5.5 所示。

第 3 步:单击“文件”工具栏中的 ,打开“保存文件”对话框,输入文件名“logo\_安全出站”,如图 5.6 所示。单击  按钮,保存程序。



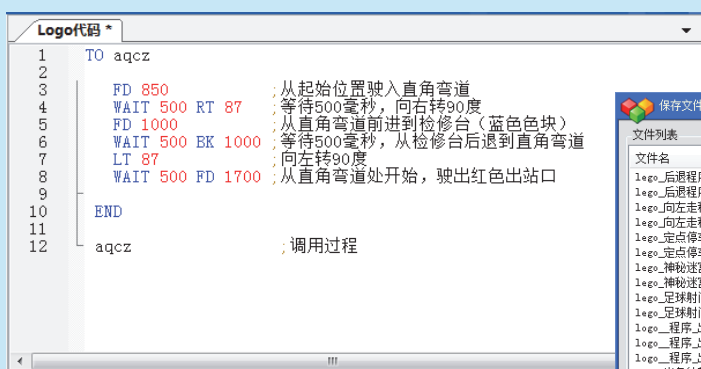


图 5.5 在程序编辑区输入程序

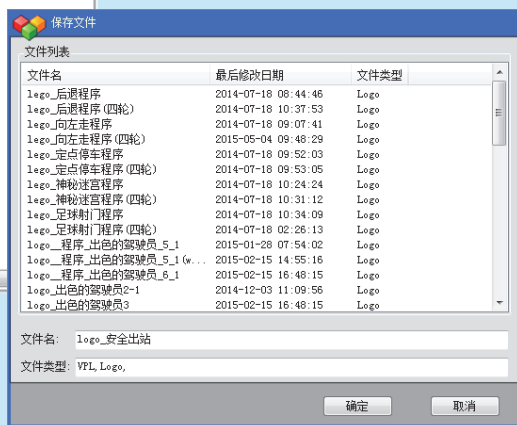


图 5.6 保存程序

### 小博士

在默认状态下, IRobotQ3D 会将文件保存到系统预设的相关文件夹中。例如, 保存机器人时会保存到“我的机器人”文件夹中, 保存仿真包时会保存到“我的仿真包”文件夹中, 保存程序时会保存到“我的程序”文件夹中。

保存程序时, 系统会先对程序进行编译, 如果程序中出现拼写或语法上的错误, 在程序编辑窗口下方将弹出错误提示信息。如图 5.7 所示, 就是将“WAIT”命令输成“WAIF”时给出的错误提示。我们可以根据提示信息进行修改, 直到把程序修改得完全无误后才允许保存退出。

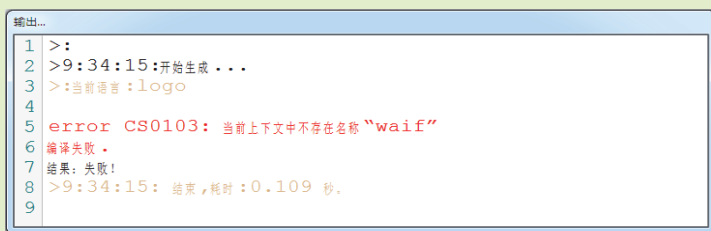


图 5.7 错误提示信息






第 4 步: 单击窗口右上方的 , 返回仿真平台, 在仿真平台单击 , 进入“快速仿真包”窗口, 单击控制区中的 , 进入“仿真导航”窗口, 按照界面上方的导航提示, 依次在左侧的列表窗格中选择: “选择任务”为场景\_安全出站、“选择机器人”为学习者号、“选择程序”为logo\_安全出站, 在 4. 导航完毕 环节的 保存仿真包: 名称框中输入“安全出站”, 如图 5.8 所示, 然后单击 保存进入 进入仿真界面。





图 5.8 导航完毕界面

**金钥匙**

1. 如果在“导航完毕”界面发现选择的机器人、控制程序有误，可以单击控制程序名称行后边的✕，移除此行选择的机器人和控制程序，然后单击 **继续添加** 重新选择。
2. 只能依次选择任务、机器人、程序，如果某选择项没完成就单击了 **下一步**，系统会给出提示，如图 5.9 所示。

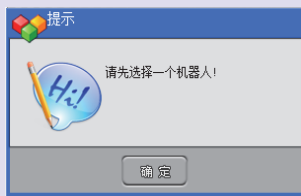


图 5.9 选择提示框







第 5 步：在仿真窗口(如图 5.10 所示)，单击仿真工具栏中的 ，开始仿真。



图 5.10 仿真窗口

### 小博士

1. 通常状况下，每一次仿真后都要单击仿真工具栏中，让机器人回归到初始状态，否则会影响下一次仿真测试。
2. 在仿真过程中，如果仿真失败或仿真效果不理想，可以单击仿真窗口工具栏中的或，重新进入编写程序窗口或搭建机器人窗口，对当前程序和机器人进行修改和调试。程序或机器人修改完毕保存退出后，会再次回到仿真窗口。



### 金钥匙

在仿真窗口中，将鼠标指针分别移动到场小地图工具栏中每一按钮上停留片刻，系统会给出提示框，说明该按钮的功能。



第6步：仿真成功后，关闭仿真窗口，退出仿真环境。

### 试试看

尝试着修改“logo-安全出站”程序，比一比谁的仿真成绩最高。



### 说说看

在 IRobotQ3D 的 LOGO 语言编程环境中编写机器人程序，需要考虑哪些实际因素？



## 三、学习实践

为“学习者号”编写的程序在 IRobotQ3D 仿真环境测试成功后，就可以将它下载到“学习者号”实物机器人的“大脑”——控制器中，使它从一台普通的“机器”变为有“智慧”的机器人，然后让它在真实的场地中“安全出站”。

给机器人的控制器下载程序，需要用数据线连接电脑和控制器，如图 5.11 所示，然后在仿真平台上执行“编译下载”命令，将调试好的程序下载到机器人的控制器中。



图 5.11 给机器人控制器下载程序





## 1. 任务


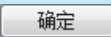
将仿真实验中编写的“logo\_安全出站”程序，下载到“学习者号”实物机器人上，并在真实场地中进行调试、运行。

## 2. 组织实施

① 6~8 个同学组成 1 个小组，在机器人实验室操作。

② 操作步骤：

第 1 步：把数据线一端连接到电脑的 USB 接口上，另一端连接到“学习者号”控制器(如图 5.12 所示)上，打开控制器电源开关。

第 2 步：在 IRobotQ3D 仿真平台打开 LOGO 程序编辑窗口，单击“文件”工具栏中的 ，弹出“打开文件”对话框，在文件列表框中选择“logo\_安全出站”，如图 5.13 所示，单击  按钮。

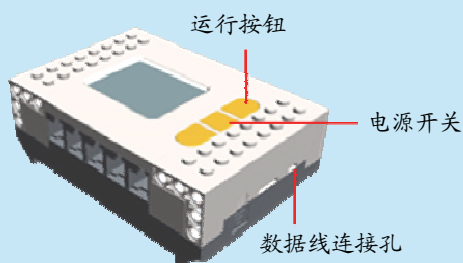


图 5.12 机器人控制器示意图

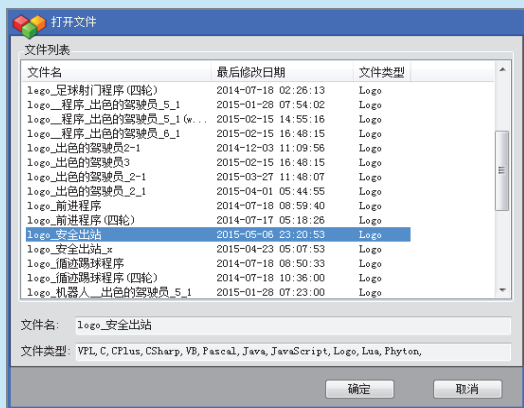



图 5.13 打开“logo\_安全出站”程序

第 3 步：在 LOGO 程序编写窗口单击“功能”工具栏中的“编译下载”按钮 ，打开“robokid-FR1 系列计算机 编译/下载”对话框，按下控制器上的“运行”按钮，开始从电脑上下载程序到“学习者号”机器人上，如图 5.14 所示。

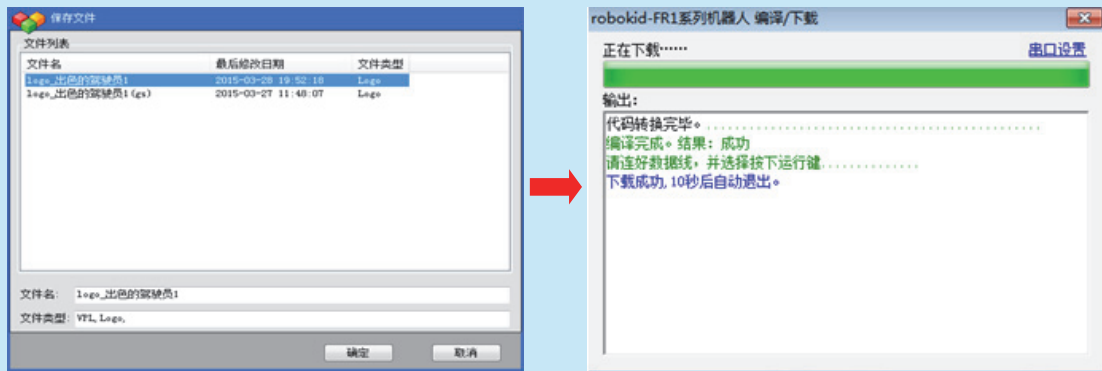


图 5.14 向实物机器人下载 LOGO 程序

### 金钥匙

如果在“robokid-FR1 系列计算机 编译/下载”对话框中出现如图 5.15 所示的内容,说明没有设置串口,这时应单击该对话框右上角的 **串口设置**,在打开的“串口设计”对话框中设置串口,如图 5.16 所示。

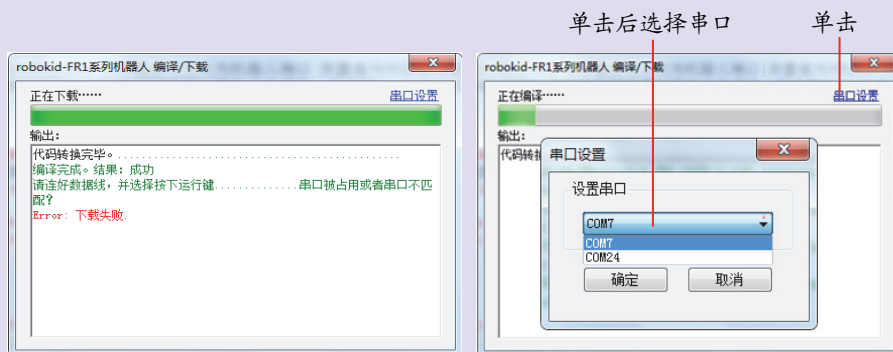


图 5.15 未设置串口的提示信息

图 5.16 设置串口

串口设置完毕后,需重新执行第 3 步才能下载程序。



第 4 步: 下载成功后,将数据线分别从计算机和机器人上拔下。

第 5 步: 参照图 5.17 所示,在光滑的平面上画出场地。

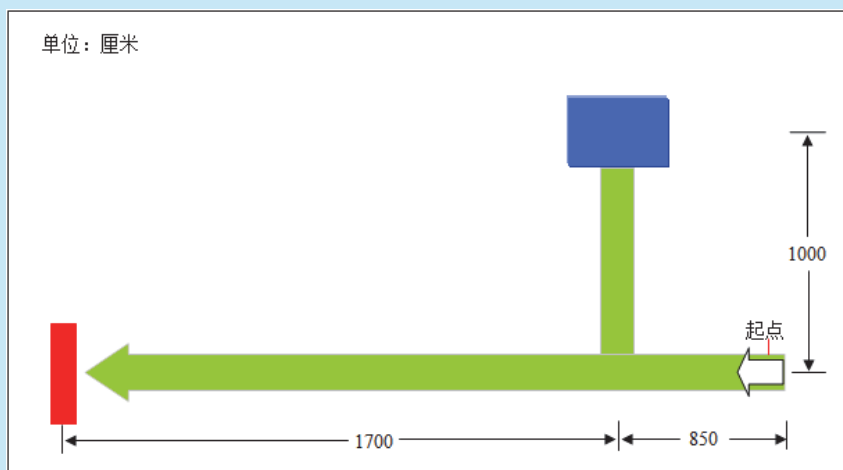
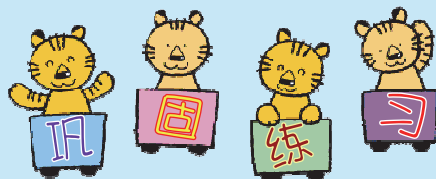


图 5.17 “安全出站”场地示意图

第 6 步: 将“学习者号”置于场地的起点位置,按下控制器上的“电源开关”键,当控制器的显示屏上出现萝卜圈图标后,按下“运行”按钮,“学习者号”机器人就会在“logo\_安全出站”程序指挥下执行安全出站任务。






1. 判断下列说法的正误。

① 机器人非常智能，不仅懂得由 0 和 1 两个符号组成的机器语言，还懂得人类用编程语言编写的程序，无需翻译就能直接运行。( )

② 程序就是程序，非常严谨，在 IRobotQ3D 的 LOGO 程序编写窗口和在 PC LOGO 语言编程窗口编写的程序都应该一模一样，不会受到实际环境的影响。( )

③ 在 IRobotQ3D 仿真平台中编写 LOGO 程序时，为了消除机器人的惯性、方便其改变运动状态，总是要用到“WAIT”命令。( )

④ 在 IRobotQ3D 仿真平台中，每一次仿真后不需要单击仿真工具栏中的进行复位，就可以继续进行仿真测试，而不会受到影响。( )

⑤ 在 IRobotQ3D 仿真平台中给实物机器人的控制器下载程序时，首先用数据线连接电脑和控制器，其次在 LOGO 程序编辑窗口执行“编译下载”命令，将调试好的程序下载到机器人的控制器中。( )

2. 在仿真平台中编写 LOGO 程序时，应该遵守的一般步骤为( )。

- A、分析任务→编写程序→仿真调试
- B、确定算法→分析任务→仿真调试→编写程序
- C、确定算法→编写程序→仿真调试
- D、分析任务→确定算法→编写程序→仿真调试

3. 在 IRobotQ3D 中根据场景任务要先搭建机器人，再编写程序，最后进行仿真调试。下列关于仿真的说法中，不正确的是( )。

- A、在仿真前一定要把机器人搭建得完美无缺并把程序编写得准确无误
- B、因为不知道搭建的机器人和编写的程序是否能完成任务，所以才要仿真调试
- C、在仿真过程中如果不能完成任务，可以适当调整机器人的结构或修改程序
- D、如果仿真任务过于复杂，可以先建一个空的程序，进入仿真后再根据实际情况编写具体的程序代码



4. 在 IRobotQ3D 仿真窗口进行仿真测试时，如果仿真失败或仿真效果不理想，就需要适当地修改程序或机器人，那么下列哪种做法最可行？( )

- A、在仿真窗口，只能直接进入 LOGO 程序编写窗口，而不能直接进入 Robokid 机器人编辑窗口，所以一般情况下只能修改程序，而不能修改机器人。



B、在仿真窗口，只能直接进入 Robokid 机器人编辑窗口，而不能直接进入 LOGO 程序编写窗口，所以一般情况下只能修改机器人，而不能修改程序。

C、关闭仿真窗口，返回到仿真平台，再进入 LOGO 程序编写窗口或 Robokid 机器人编辑窗口。

D、在仿真窗口，通过单击仿真工具栏中的  或 ，直接进入 LOGO 程序编写窗口或 Robokid 机器人编辑窗口。

5. 对比 IRobotQ3D 的 LOGO 程序编写窗口与第 2 单元的 LOGO 语言编程窗口(见第 2 单元的图 2.2)，指出它们的相同和不同之处。

6. 说一说 IRobotQ3D 仿真窗口工具栏中各个按钮的功能？你在仿真过程中都使用过哪些按钮？说一说它们都适合在什么情况下使用？

7. 在 IRobotQ3D 在线仿真平台编写 LOGO 程序，指挥“学习者号”完成“新手入门\_前进”仿真任务。

8. 在 IRobotQ3D 在线仿真平台编写 LOGO 程序，指挥“学习者号”完成“新手入门\_后退”仿真任务。

9. 在 IRobotQ3D 在线仿真平台编写 LOGO 程序，指挥“学习者号”完成“新手入门\_向左走”仿真任务。

10. 在 IRobotQ3D 在线仿真平台编写 LOGO 程序，指挥“学习者号”完成“新手入门\_向右走”仿真任务。

11. 在 IRobotQ3D 中编写一段 LOGO 程序，再将其下载到实物机器人“学习者号”上，让它在实际环境中太空漫步吧！下面是“太空漫步”的算法(仅供参考)。

- ① 前进 10 步；
- ② 后退 10 步；
- ③ 左转 90 度；
- ④ 前进 10 步；
- ⑤ 后退 10 步；
- ⑥ 右转 90 度；
- ⑦ 停止。





## 第6单元 新手上路



“学习者号”在程序“指挥”下前进、后退指定的距离或停止规定的时间，完成了安全出站任务。如果要它到道路上行驶，就必须能应对道路上错综复杂的交通状况，这仅靠我们已经给它输入的那点“智慧”是不够的，它必须学会自己收集信息并做出判断的本领。

让“学习者号”具有收集信息并做出判断的本领并不难，只要为它配置类似人类感觉器官的传感器，并编写相应的程序指令就可以了。

传感器的种类很多，从这一单元起，我们通过为“学习者号”机器人配置常用的几类传感器，体验“学习者号”由简单机器人逐步变为智能机器人的过程。

### 一、超声波传感器

超声波传感器是机器人常用的一种传感器，它是声敏传感器中的一种，类似于蝙蝠的耳朵。图 6.1 所示是“学习者号”使用的超声波传感器。

超声波传感器常用于在有效范围检测机器人到障碍物的距离。它工作时会发射一条黑白相间的光线，在一定范围内，如果没有障碍物，发射出去的超声波会因为传播距离渐远而逐渐减弱，最后消失；如果有障碍物，超声波遇到障碍物会被反射到传感器的超声探头，传感器检测到这一信号，就可以确定与障碍物的距离，并将信号传送给控制器，控制器根据信号调用程序指令，指挥机器人完成相应动作。



图 6.1 超声波传感器

### 二、在 IRobotQ3D 中给机器人添加超声波传感器

在 IRobotQ3D 中给机器人添加传感器的方法，与添加其他模型的方法类似。在机器人编辑窗口的模型库中单击“模型”标签，在打开的“模型”面板中单击 **传感器**，列表窗格中就显示出各种传感器，其中第二排第



图 6.2 IRobotQ3D 中的超声波传感器

1 个就是超声波传感器，如图 6.2 所示。

在 IRobotQ3D 虚拟环境中，超声波传感器的检测范围为 1~3000，当障碍物与机器人的距离超过检测范围时将检测不到。

### 1. 构建超声波传感器组件

在 IRobotQ3D 中，安装到机器人上的超声波传感器应该是一个组件，通常由支架、连接件、传感器等零部件组成，我们应该先将这些零部件构建成一个组件（即第 4 单元中提到的模板），再将它安装到机器人上。



利用表 6.1 中的零部件构建超声波传感器模板，以“超声波传感器-1”为备注添加到 IRobotQ3D 的模板中。

表 6.1 组成超声波传感器模板所需零件清单

名称	图片	数量	作用
超声波传感器		1	检测公交站台广告牌及与公交站台广告牌的距离
方杆 3 孔		1	连接超声波传感器与 90° 连接件
90° 连接件		1	连接方杆 3 孔与厚连杆 3 孔
厚连杆 3 孔		1	连接 90° 连接件与 2 个双接口
十字转轴		4	用于方杆 3 孔与 90° 连接件、90° 连接件与厚连杆 3 孔、厚连杆 3 孔与双接口的连接
双接口		2	连接超声波传感器与控制器

第 1 步：启动 IRobotQ3D，在仿真平台中打开 Robokid 机器人编辑窗口。

第 2 步：单击模型库“模型”面板中的 **安装块**，在打开的安装块模型列表中分别选择“方杆 3 孔”“十字转轴”和“90°连接件”，依次添加到编辑区，如图 6.3 所示。

第 3 步：在编辑区单击“十字转轴”出现橘色安装点后，再单击右侧圆形安装点使其变为粉色，接着单击

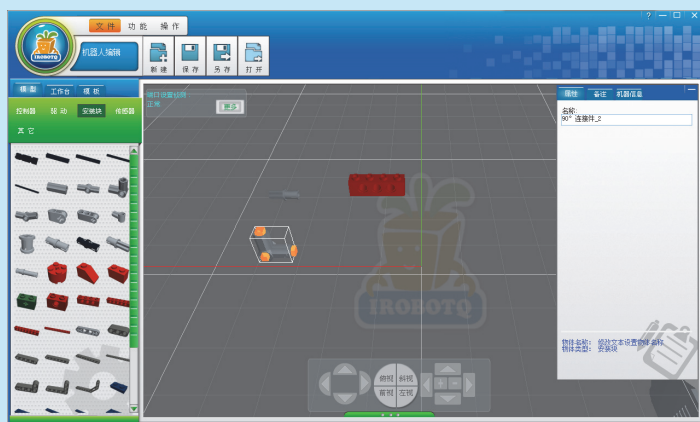


图 6.3 添加“方杆 3 孔”“十字转轴”和“90° 连接件”



“方杆 3 孔”中间的孔形安装点，将“十字转轴”安装到“方杆 3 孔”上，然后单击“90°连接件”，再单击右侧十字孔形安装点使其变为粉色，接着单击“十字转轴”的十字形安装点，将“90°连接件”安装到“十字转轴”上；最后按 2 次空格键，将“90°连接件”旋转 180 度安装到“方杆 3 孔”上，如图 6.4 所示。

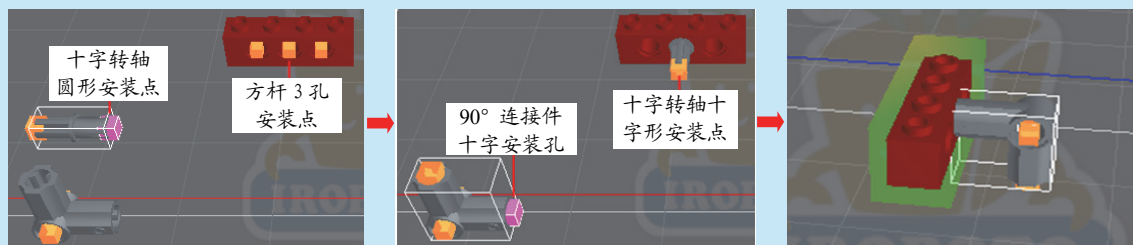


图 6.4 安装“方杆 3 孔”“十字转轴”和“90° 连接件”

### 金钥匙

在安装过程中，要寻找模型中形状一致的孔与柱相连接，一般是圆形的孔与圆形的柱连接，棱形的孔与棱形的柱连接。



第 4 步：在“安装块”模型列表中，分别选择 1 次“厚连杆 3 孔”和 3 次“十字转轴”，依次添加到编辑区。

第 5 步：参照图 6.5，在编辑区将其中的 1 个“十字转轴”圆形安装点，安装到“厚连杆 3 孔”左前方下数第一个孔形安装点内；再将另外 2 个“十字转轴”，依次安装到“厚连杆 3 孔”右侧另外 2 个孔形安装点上。

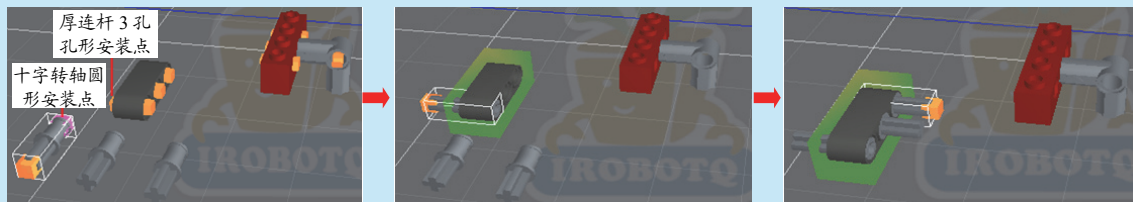




图 6.5 安装“厚连杆 3 孔”和“十字转轴”

### 金钥匙

如果某个模型安错了位置，可用以下方法解决：

1. 单击“操作”工具栏中的“撤销”按钮，然后重新安装。
2. 先拆卸模型，再重新安装。具体做法为：先选择该模型，单击“操作”工具栏中的“结构拆卸”按钮，再将鼠标指针移到模型上，按下鼠标左键拖动到脱离安装位置，然后重新安装。



第 6 步：参照图 6.6，在编辑区将“厚连杆 3 孔”左前方“十字转轴”的十字形安装点，安装到“90°连接件”下方的十字形安装孔内。



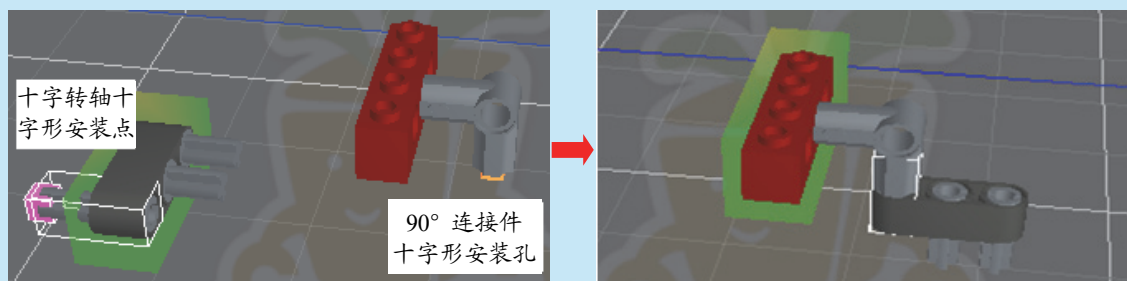


图 6.6 安装“90°连接件”和“厚连杆 3 孔”

### 金钥匙

利用“90°连接件”，可以使两个连接件在角度上产生较大的改变。



第 7 步：在“安装块”模型列表中分别选择 2 次“双接口”，依次添加到编辑区。

第 8 步：参照图 6.7，在编辑区分别将 2 个“双接口”前方十字形安装孔，安装到“厚连杆 3 孔”右下方 2 个“十字转轴”十字形安装点上。

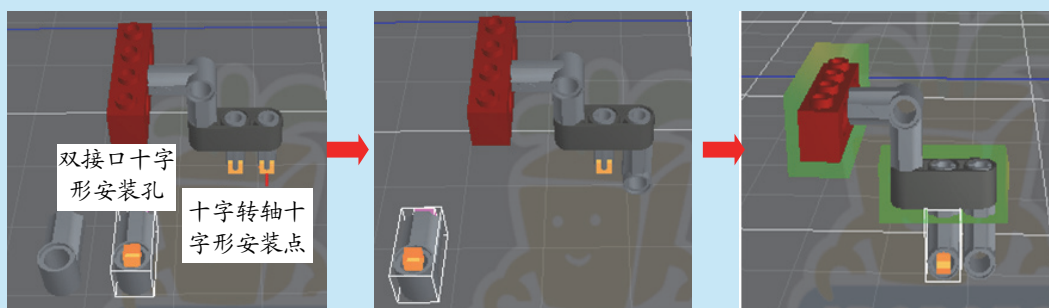


图 6.7 安装“双接口”和“厚连杆 3 孔”

第 9 步：单击“模型”面板中的 **传感器**，在“传感器”列表中选择“超声波传感器”添加到编辑区，再按 5 次空格键，使超声波传感器旋转至显示出凹形安装点，然后将超声波传感器左下方第三行第 1 列的安装点安装到“方杆 3 孔”上。如图 6.8 所示。

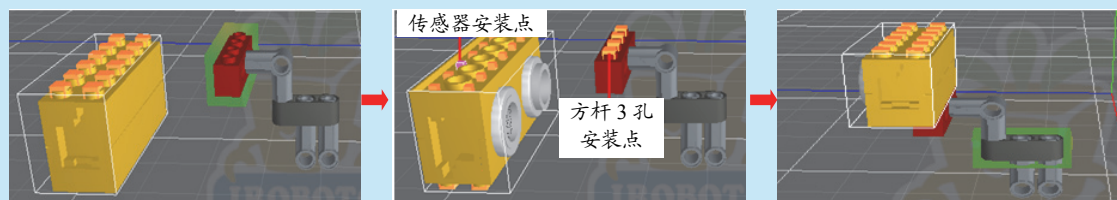


图 6.8 安装超声波传感器到“方杆 3 孔”

第 10 步：将组合完成的部件以“超声波传感器-1”为模板名，保存到“我的模板”中，如图 6.9 所示。





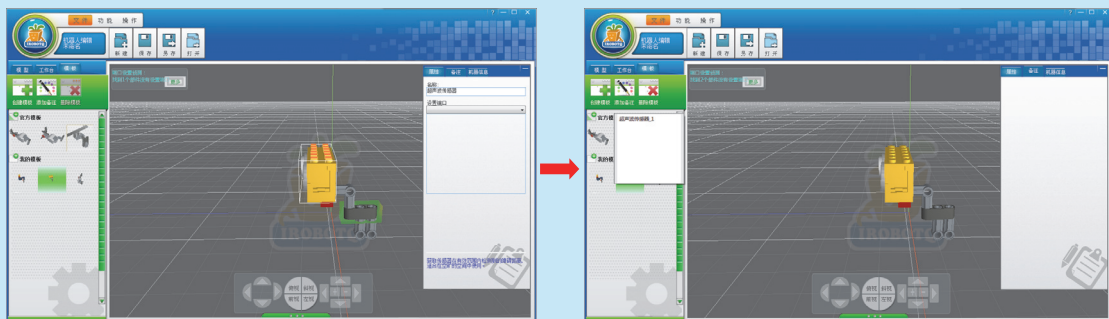


图 6.9 将“超声波传感器-1”添加为模板

**试试看**

1. 分别将模型库“传感器”列表窗格中的各种传感器，拖动到机器人编辑窗口的编辑区，通过按空格键改变其视角，仔细观察每种传感器检测方向的所在位置。
2. 打开 Robokid 机器人编辑窗口，添加“超声波传感器-1”模板，试着将超声波传感器拆卸下来，换上其他传感器。

**2. 给机器人添加超声波传感器组件**

完成了超声波传感器模板的搭建后，再搭建一个新的“学习者1号”机器人就容易多了，我们只要打开 IRobotQ3D 的 Robokid 机器人编辑窗口，在“学习者号”的基础上将“超声波传感器-1”组装上去就可以了。

**做一做**

在“学习者号”机器人的基础上添加一个超声波传感器部件，将其另存为“学习者1号”。

第1步：登录 IRobotQ3D，在仿真平台中打开 Robokid 机器人编辑窗口。


第2步：单击“文件”工具栏中的 ，打开“选择机器人”对话框，在机器人列表窗格选择“学习者号”，如图 6.10 所示，单击 **确定** 按钮，“学习者号”显示在 Robokid 机器人编辑窗口的编辑区中。



图 6.10 “选择机器人”对话框



第 3 步：在编辑区单击“学习者号”，按下鼠标右键向右拖动，显示左电机部件，再单击模型库“模型”面板中的**安装块**，在打开的模型列表中选择 2 次“单边加长转轴”，依次将它们添加到编辑区。

第 4 步：在编辑区单击其中的一个“单边加长转轴”，按 2 次空格键，使“单边加长转轴”旋转 180 度显示短轴安装点，将“单边加长转轴”的短轴安装点，安装到“学习者号”右侧电机上的“厚连杆 3×5L 型”左数第 2 个安装孔内，如图 6.11 所示。用相同的方法，将另一个“单边加长转轴”的短轴，安装到此“厚连杆 3×5 孔 L 型”左数第 1 个安装孔内。

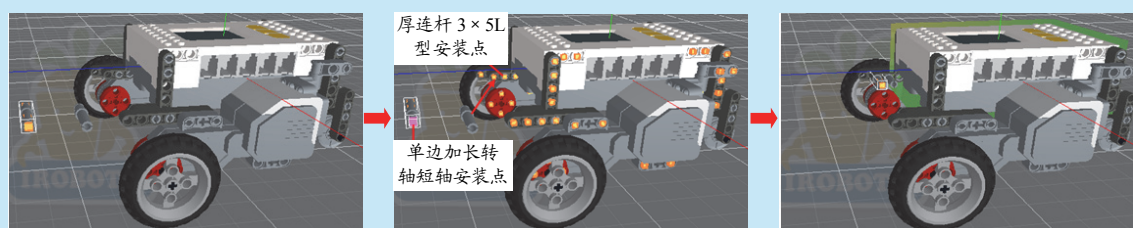


图 6.11 安装“单边加长转轴”

第 5 步：单击模型库中的“模板”标签，在打开的列表窗格中，单击“我的模板”中的“超声波传感器-1”，把它添加到编辑区“学习者号”左侧，接着按 5 次空格键，使超声波传感器旋转至检测口向右；再将“超声波传感器-1”左下方的“双接口”安装孔，与机器人右侧“厚连杆 3×5L 型”右边的“单边加长转轴”的长轴安装点相连，最后按 2 次空格键，使“超声波传感器-1”旋转 180 度安装到机器人右前方，如图 6.12 所示。

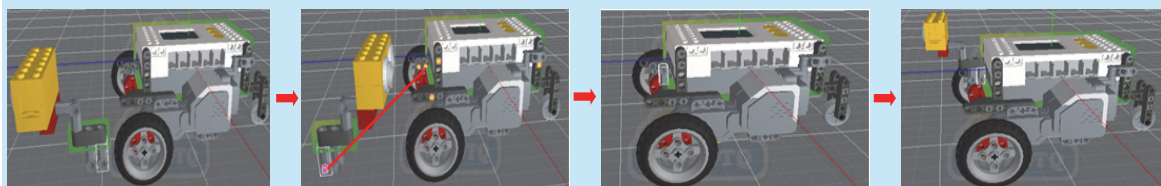


图 6.12 安装“超声波传感器-1”模板

第 6 步：在编辑区先单击机器人，按下鼠标右键向上拖动，显示机器人底部，再单击连接“厚连杆 3 孔”和“90°连接件”的“十字转轴”，接着单击窗口上方的**操作**标签，在“操作”工具栏的**旋转**中输入 X 的值“-45”，然后按回车键确认，“超声波传感器-1”顺时针旋转 45°，如图 6.13 所示。



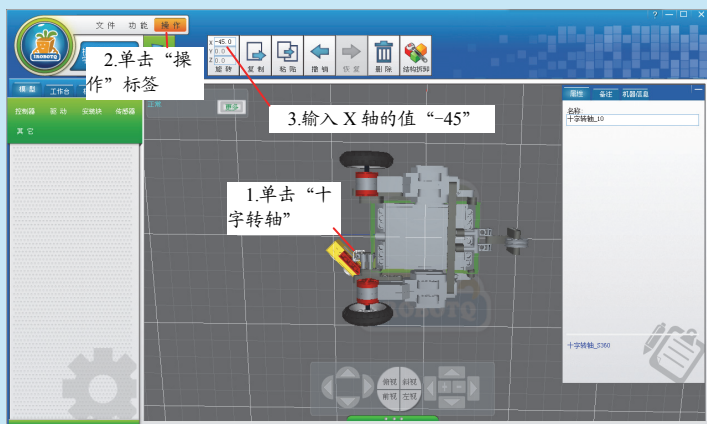



图 6.13 旋转超声波传感器

**金钥匙**

在 RoboDK 机器人编辑窗口中搭建机器人时，如果要精确旋转某个部件，可以选择该部件后，利用“操作”工具栏中的 ，输入具体的角度值来实现。

**小博士**

在下面的“做一做”中，要求机器人能辨认站台上的广告牌。理论上为了让机器人辨认广告牌，超声波传感器的检测方向应该与它前进方向垂直。但是由于机器人有惯性，判断时需要有一个提前的预知，所以实际安装时，超声波传感器的检测方向需要依据机器人的速度，与正前方设置一定角度。这个角度需要调节，机器人速度越快，传感器检测方向与正前方的夹角越小。



第 7 步：将鼠标指针移到机器人上，按下鼠标右键向下拖动，显示超声波传感器后单击超声波传感器，然后单击属性面板中“属性”标签，在“设置端口”列表中选择端口“31”，如图 6.14 所示。

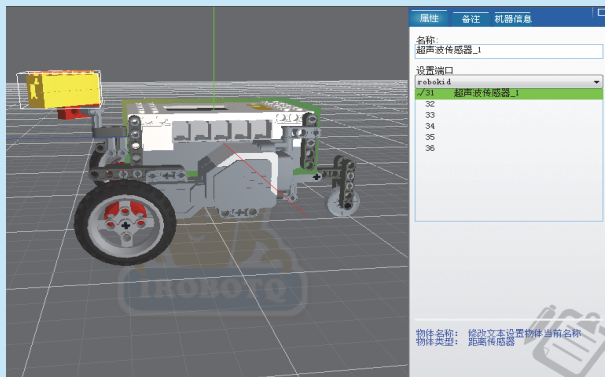


图 6.14 设置超声波传感器的属性



### 小博士

端口是计算机(这里指控制器中的单片机)与外界通讯交流的接口。为了让控制器能明确地控制传感器,通常必须给每个传感器分配唯一的端口号。在 RobotQ3D 的 Robokid 机器人编辑窗口中,传感器的可用端口范围为 31~36。

传感器端口变量可以表示为“:IO+端口号”,在本例中超声波传感器的端口号设为 31,所以它的端口变量可表示为“:IO31”。

如果没有给传感器设置端口,在保存机器人时会弹出对话框进行提示。



第 8 步:单击“文件”工具栏中的,打开“保存机器人”对话框,将文件名“学习者号”修改为“学习者 1 号”,单击按钮,保存机器人。

### 试试看

1. 调整“学习者 1 号”上超声波传感器的检测面朝向。
2. 给“学习者号”加装不同的传感器,并相互交流为什么安装在这个位置?安装这种传感器的目的是什么?



## 三、仿真体验使用超声波传感器

由于“学习者 1 号”上配备了超声波传感器,因此它具备了“感知”道路附近特定物体的能力,利用它的这一功能设计相应程序,可以让“学习者 1 号”完成很多任务。



图 6.15 是 IRobotQ3D 提供的编号为“精准停靠”的仿真场地,让“学习者 1 号”先从出站口沿箭头方向驶上公路后右转,沿着公路向前行驶到公交站台(此处有广告牌)接送乘客,然后再继续前进。

### 1. 任务分析

怎样才能让“学习者 1 号”准确无误地停靠到公交站台呢?“学习者 1 号”已经安装了超声波传感器,如果在公交站台上选择一个标识物作为停靠标志(本例取公交站台前高出地面的广告牌),让“学习者 1 号”“感觉到”这个停靠的信息,据此做出判断,就能执行相应命令完成这个任务。

“学习者 1 号”具有了感知能力,如果我们再赋予它判断能力——在程序设计中应用条件判断命令,使超声波传感器在指定范围内(250 步内)检测到站台前广告牌,就停止运行靠站接客,否则继续前进。



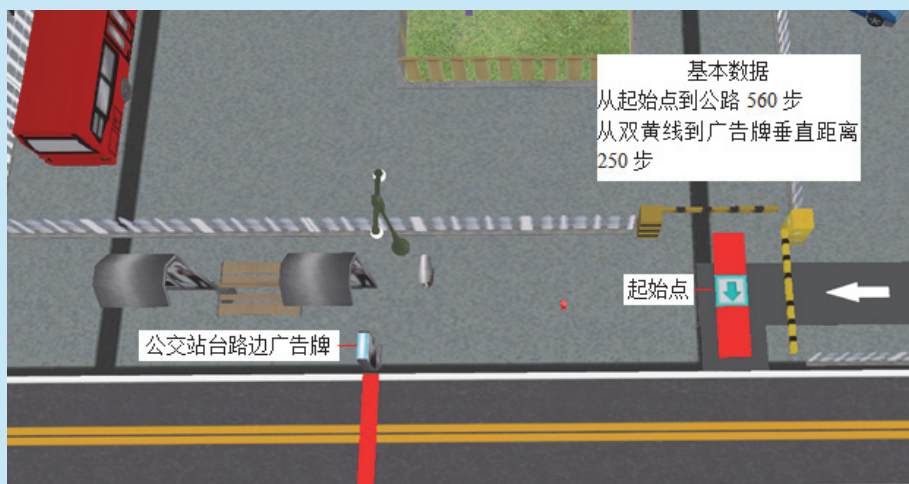


图 6.15 “精准停靠”仿真场地示意图


## 2. 算法描述

实现“精准停靠”的过程可以用流程图表示，如图6.16所示。

由流程图可以看出，要实现“精准停靠”，“学习者1号”应一边前进一边判断，在 LOGO 语言编程中，实现这一功能需要调用“尾递归”。

## 3. 编写程序与仿真调试

第1步：登录 IRobotQ3D，由仿真大厅进入仿真平台，打开 LOGO 程序编写窗口。

第2步：单击“文件”工具栏中的 ，单击程序编辑区，参照图 6.17 输入程序。

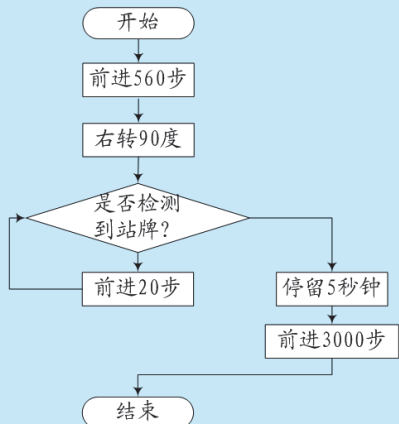


图 6.16 “精准停靠”流程图



```
Logo代码
1  TO jztk ; 机器人寻找停车站牌
2      fd 20 ; 前进
3
4      ; 如果超声传感器检测到站牌，则停靠5秒后继续前进
5      ; 否则继续寻找停车站牌
6      if :I031<250 then
7          [
8              fd 400
9              wait 5000
10             fd 3000
11         ]
12      else
13          [
14              jztk ; 尾递归调用
15          ]
16      END
17
18  ; 机器人行至公路上后右转弯
19  fd 560 wait 500
20  rt 87 wait 500
21
22  ; 调用过程
23  jztk
```

图 6.17 在程序编辑区输入程序

第3步：单击“文件”工具栏中的 ，在打开的“保存文件”对话框中输入文件名“logo\_精准停靠”，单击  按钮，保存程序。

第4步：关闭 LOGO 程序编写窗口，由仿真平台进入快速仿真包窗口，单击控




制区中的，按照界面上方的导航提示，依次在任务列表窗格中选择“选择的任务”为**精准停靠**，“选择的机器人”为**学习者1号**、“选择的程序”为**logo\_精准停靠**，在**4. 导航完毕**环节的**保存仿真包**：名称框中输入“精准停靠”，如图 6.18 所示，然后单击**保存进入**进入仿真窗口。




图 6.18 导航完毕界面

### 金钥匙

在程序的调试过程中，由于受仿真环境的影响较大，因此通常采用一边测试一边修改的方法。为了提高测试效率，可以一段一段地测试，对不参与测试的语句标注上注释语句符号“；”。



第 5 步：单击仿真工具栏中的，开始仿真。

### 说说看

仔细分析“logo-精准停靠”程序，说一说“学习者 1 号”是如何实现一边检测公交站台广告牌，一边行走的？



### 知识窗

#### 用流程图表示算法

在程序设计中，算法既可以用自然语言描述，又可以用流程图表示。

流程图也称为程序框图，是用带箭头的流程线把各种表示不同含义的框连接起来的一种表示算法的形式。通常情况下，使用流程图可以更直观更清晰地设计和描述算法。表 6.2 就是一些常用的程序流程图符号及其表达的意义。





表 6.2 常用的程序流程图符号

名 称	流程图符号	意 义
起止框		表示程序的开始或结束
输入或输出框		表示数据的输入或输出操作
判断框		表示根据条件进行判断，选择程序下一步的执行方向
处理框		表示处理方法
箭头线		表示程序流程的走向

在上一单元“安全出站”的程序设计中，由于任务相对简单，所以在确定算法时我们用自然语言进行描述，如果用流程图来描述该任务，结果如图 6.19 所示，看起来还没有用语言描述简洁。

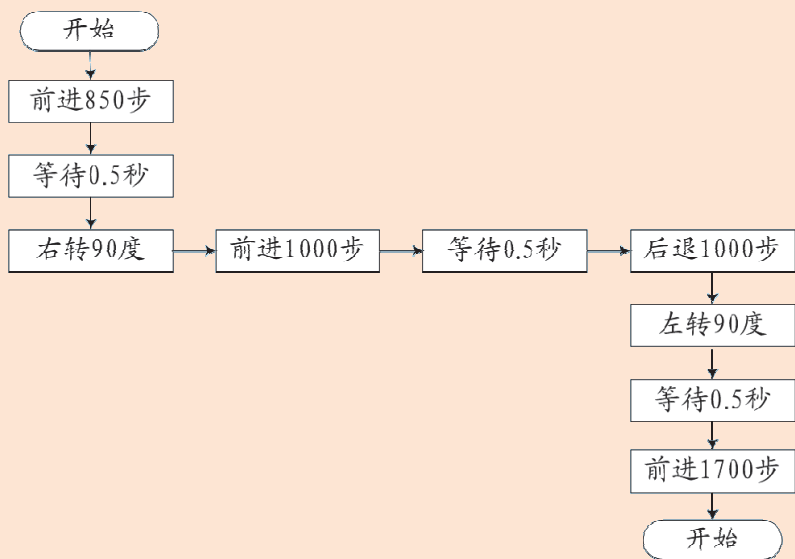


图 6.19 “安全出站”流程图

但当任务逐渐复杂时，再使用自然语言描述算法就会变得相当烦琐。利用流程图描述算法可以使描述变得简洁，针对“精准停靠”任务的算法描述就是一例。



### 试试看

修改程序“logo\_精准停靠”中的命令参数，提高仿真分值。





## 四、学习实践

“学习者 1 号”装备了超声波传感器，并为它设计了具有判断能力的程序，经仿真环境测试成功，现在可以在真实的“学习者号”上进行“感觉”测试。

### 1. 任务

- ① 给实物机器人“学习者号”添加超声波传感器。
- ② 将仿真实验的“logo\_精准停靠”程序下载到“学习者号”上，并在真实场地中进行调试、运行。

### 2. 组织实施：

- ① 6~8 个同学组成 1 个小组，在机器人实验室操作。

#### ② 操作步骤：

第 1 步：参照本单元第 2 个“做一做”，给“学习者号”添加编号为“CG-CSB-01”的超声波传感器。

第 2 步：将“logo\_精准停靠”程序从电脑下载到“学习者号”上。

第 3 步：参考图 6.15，在光滑的平面上画出场地，如图 6.20 所示。

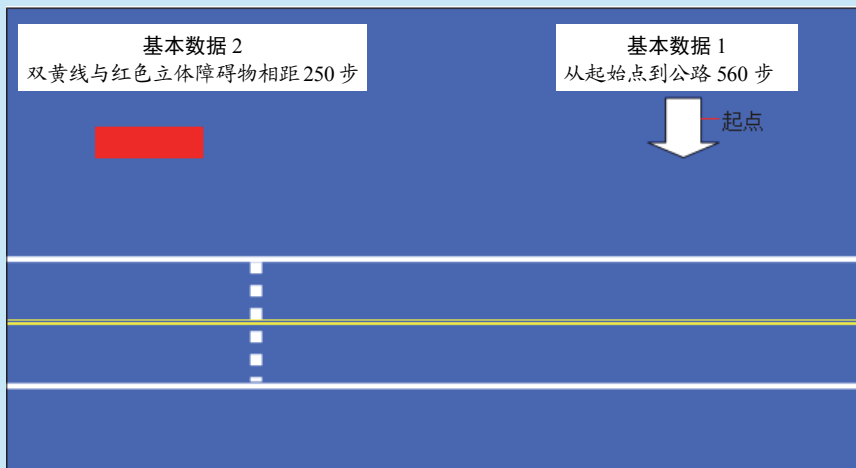
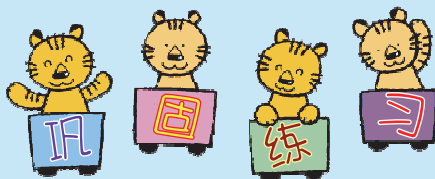


图 6.20 “精确停靠”场地示意图


第 4 步：在“精准停靠”场地调试、运行“学习者号”。



1. 判断下列说法的正误。

① 小明说：“搭建机器人时，不需要关注各个零部件的形状，因为它们都是标准件，随便组合，只要能安装到一起就行！”（ ）


② 超声波传感器在一定的范围内，不仅可以检测障碍物，而且可以检测与障碍物之间的距离。（ ）

③ 在 IRobotQ3D 的 Robokid 机器人编辑窗口，利用“操作”工具栏中的 ，可以实现对模型的精确旋转。（ ）

④ “logo\_精准停靠”程序是靠 4 条基本命令和一个过程调用来实现的。（ ）

⑤ 所有程序的算法都适合用流程图表示。（ ）

2. 在搭建机器人时，下列哪种方法无法实现对选定模型的旋转？（ ）

- A、在“操作”工具栏的  中输入具体的角度值  
B、按键盘上的空格键  
C、按住鼠标右键拖动  
D、按住鼠标左键拖动

3. 在 IRobotQ3D 中编写 LOGO 程序时，能正确表示 Robokid 机器人端口变量的选项是（ ）。

- A、:IO 端口号  
B、: IO 端口号  
C、;IO 端口号  
D、; IO 端口号

4. 在 IRobotQ3D 的 Robokid 机器人编辑窗口中给机器人安装传感器时，必须给每个传感器设置端口，下面给出的端口号范围中正确的是（ ）。

- A、1~29  
B、30~60  
C、60~66  
D、30~36

5. 说一说为什么要给机器人安装各种各样的传感器？

6. 在模型库的“传感器”列表中，将鼠标指针分别移动到每种传感器图标上停留片刻，通过弹出的提示框了解各种传感器的名称及功能，并观察各个传感器的检测方向。

7. 为什么通常状况下，安装传感器时不是直接安装到机器人上，而是需要先搭建由一个支架、连接件、传感器等元件组成的组件，再对它们进行安装？

8. 相互交流一下，在搭建机器人的过程中，如何快速、准确地找到各个零部件的安装点？

9. 用自然语言描述算法和用流程图表示算法各有什么优缺点？



## 阅读材料

### 机器人常用的几种传感器

机器人常用的传感器种类很多，以 IRobotQ3D 仿真机器人环境中的 IROBOTQ 类型机器人(参见图 4.1)为例，IRobotQ3D 为这种机器人提供了 8 种传感器模块，它们分别是距离传感器、障碍传感器、灰度传感器、光线传感器、接触传感器、指南针传感器、海拔高度计和红外探测器。这里简单介绍其中 4 种传感器的作用和功能。

① 距离传感器：用于检测机器人某个方向的前方有无物体及与物体的距离。其检测距离是 0~3000，当距离传感器的返回值为 0 到 3000 之间时，表明此传感器的前方(3000 范围内)有物体，具体的数值表示传感器与物体之间的距离。与物体距离超过 3000 时，返回值为 3000。

距离传感器在工作时会发射出一条蓝白相间的射线，指示其检测的方向和检测范围。射线照射到物体上表示检测到了此物体，射线会被物体截断。

② 光线传感器，用于检测其前方的光线强度。理想条件下，光线传感器在 0 距离、0 角度检测 1 支蜡烛时，其返回值为 1000，这也被称为 1 坎德拉发光强度。光线传感器距离光源越近，返回值越大，反之越小；与光源的夹角越小，返回值越大，反之越小。

安装光线传感器时，应与要检测到的光源中心保持在同一水平面上，其前方不要有遮挡物，为扩大检测的范围，可以在不同的方向安装多个光线传感器。

③ 指南针传感器：用于获取方向，返回其前方(指南针上标有“N”的方向)与场景正北方向的夹角，返回值为 0~359 度。指南针传感器常用于使机器人转到某个方向，或使机器人沿着某个方向行进。使用指南针传感器时需要水平安装，且指针面朝上，否则可能无法正常工作。

④ 海拔高度计：返回传感器相对于海平面的垂直距离，返回值为整型。在系统中海平面是 Y 轴坐标为 0 的 XZ 平面，海拔高度计的位置高于海平面时返回正值，低于海平面时返回负值。海拔高度计常用于检测机器人行进过程中的高度变化。

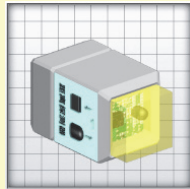


图 6.21 距离传感器

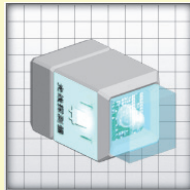


图 6.22 光线传感器



图 6.23 指南针传感器

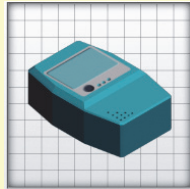


图 6.24 海拔高度计





## 第7单元 小试牛刀



机动车在道路上需要按照交通标识线行驶。“学习者 1 号”配备“超声波传感器”后能辨识障碍物及与它的距离，但还不能识别道路上的交通标识线。“学习者 1 号”要想上路行驶，还得为它配备能辨识标识线的传感器。交通标识线是不同颜色的线条，能用于教学机器人辨识颜色的传感器种类很多，灰度传感器就是其中一种。

### 一、灰度传感器

灰度传感器是模拟传感器，它是光敏传感器的一种，有一只发光二极管和一只光敏电阻安装在同一面上，用来检测物体表面灰度的变化，图 7.1 所示是“学习者 1 号”使用的灰度传感器。



图 7.1 灰度传感器

不同颜色的表面对光的反射程度不同，灰度传感器利用光敏电阻对不同检测面返回的光产生的电阻值不同这一原理检测颜色深浅。

### 二、在 IRobotQ3D 中给机器人添加灰度传感器

#### 1. 构建灰度传感器组件

与安装多数传感器一样，在 IRobotQ3D 中给机器人安装灰度传感器，也需要先用支架、连接件、传感器等零部件构成组件，再将它安装到机器人上。



**做一做** 安装到机器人上的灰度传感器需要表 7.1 所列零部件。利用这些零部件构建一个灰度传感器组件，以“灰度传感器-1”为备注添加到 IRobotQ3D 的模板中。

表 7.1 安装灰度传感器所需零部件清单

名称	图片	数量	作用
灰度传感器		1	检测双黄线灰度值
方杆 3 孔		5	用于板 6×8 柱与厚连杆 3×5 孔 L 型，灰度传感器与板 6×8 柱的连接
板 6×8 柱		1	连接方杆 3 孔及方杆 3 孔组

名称	图片	数量	作用
厚连杆 3×5 孔 L 型		1	连接方杆 3 孔与控制器
转轴		2	连接厚连杆 3×5 孔 L 型与方杆 3 孔

第 1 步：登录 IRobotQ3D，在仿真平台中打开 Robokid 机器人编辑窗口，在模型库“模型”面板“安装块”模型列表中，分别选择 1 次“厚连杆 3×5 孔 L 型”、2 次“转轴”和 5 次“方杆 3 孔”，依次添加到编辑区，如图 7.2 所示。

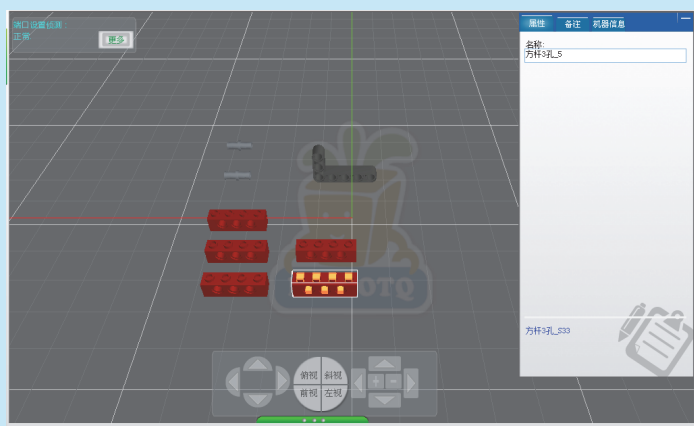


图 7.2 添加“厚连杆 3×5 孔 L 型”、“转轴”和“方杆 3 孔”

第 2 步：在编辑区先单击“厚连杆 3×5 孔 L 型”，再按 5 次空格键，使其旋转至 L 型短轴向下；接着分别将 2 个“转轴”依次安装到“厚连杆 3×5 孔 L 型”短轴上第 1 和第 3 个安装孔内，如图 7.3 和图 7.4 所示。

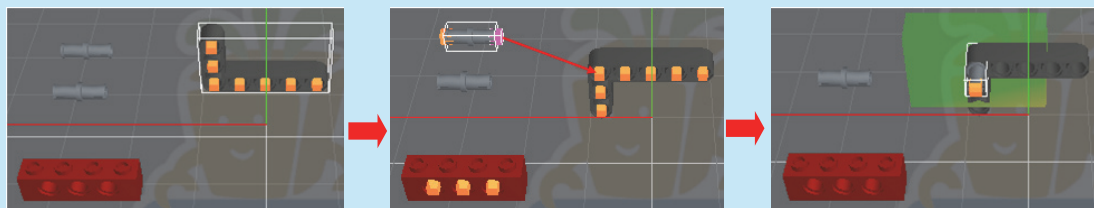


图 7.3 将第 1 个“转轴”安装到“厚连杆 3×5 孔 L 型”上

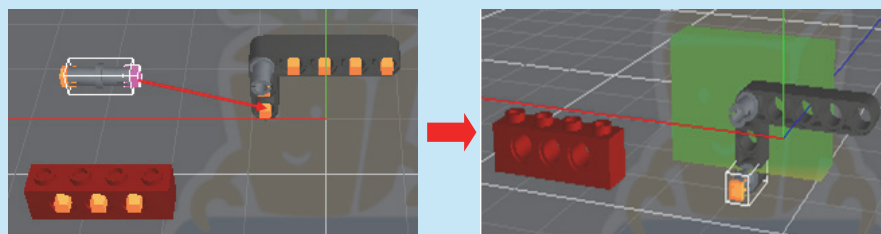


图 7.4 将第 2 个“转轴”安装到“厚连杆 3×5 孔 L 型”上





第3步：参照图 7.5，将其中一个“方杆 3 孔”右数第 1 个孔状安装点，安装到“厚连杆 3×5 孔 L 型”短轴下数第 3 个安装点上，再按 1 次空格键，使“方杆 3 孔”旋转 90 度，纵向安装在“厚连杆 3×5 孔 L 型”上。

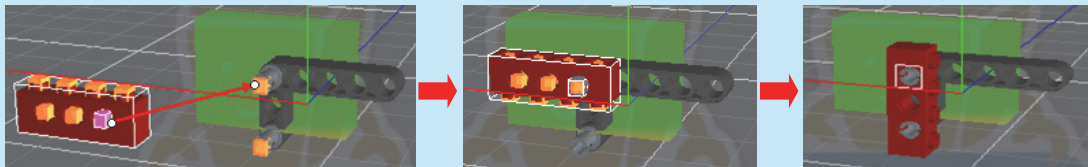


图 7.5 将“方杆 3 孔”安装到“厚连杆 3×5 孔 L 型”上

第4步：参照图 7.6，先单击左侧一个“方杆 3 孔”，按 5 次空格键，使其旋转至底部凸状安装点向上，再将这个“方杆 3 孔”右数第 1 个安装点，安装到它右侧“方杆 3 孔”右数第 1 个安装点上，让它们横向叠加在一起。



图 7.6 将 2 个“方杆 3 孔”安装在一起

第5步：参照第4步和图 7.7，将另外 2 个“方杆 3 孔”横向叠加在一起。



图 7.7 将另外 2 个“方杆 3 孔”安装在一起

第6点：在编辑区先单击“厚连杆 3×5 孔 L 型”，按下鼠标右键缓缓向右拖动，显示“方杆 3 孔”有 4 个凸状安装点的一侧，再在“安装块”模型列表中选择“板 6×8 柱”添加到编辑区，接着将“板 6×8 柱”右上方第 3 排第 3 列的安装点，与“方杆 3 孔”上数第 1 个凸状安装点相连，然后按 1 次空格键，使“板 6×8 柱”旋转 90 度，横向安装在“方杆 3 孔”上，如图 7.8 所示。

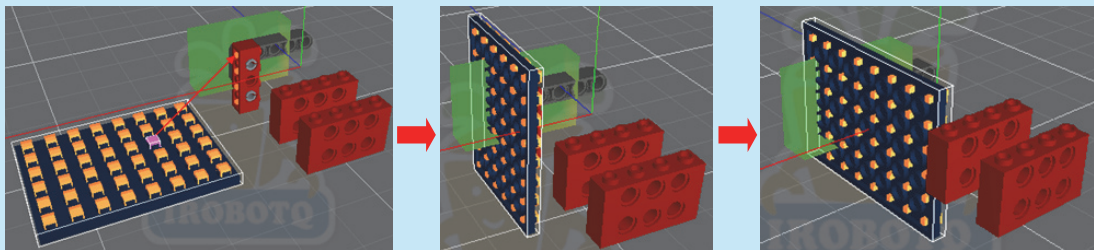


图 7.8 安装“板 6×8 柱”



第7步：参照图 7.9，将一组“方杆 3 孔”左侧第一个安装点，与“板 6×8 柱”右上方第 3 排第 1 列的安装点相连，再按 1 次空格键，使这组“方杆 3 孔”旋转 90 度，纵向安装在“板 6×8 柱”上。

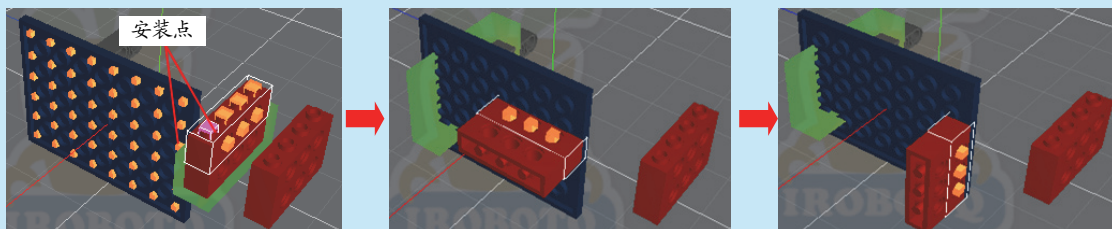


图 7.9 安装一组“方杆 3 孔”

第8步：参照第7步和图 7.10，把另外一组“方杆 3 孔”纵向安装在“板 6×8 柱”右上方第 3 排第 2 列上。

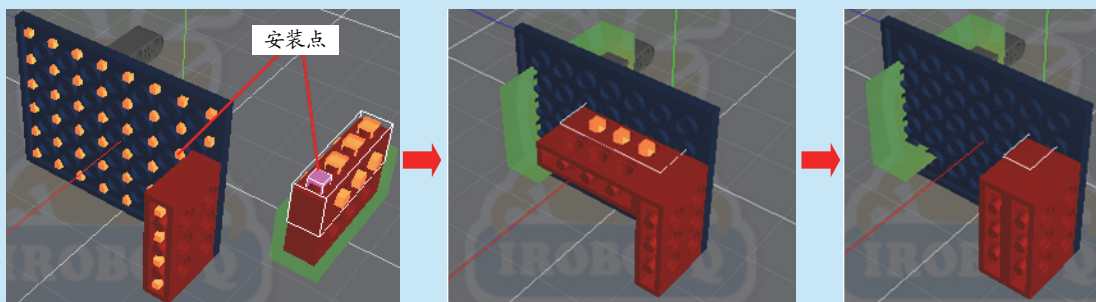


图 7.10 安装另一组“方杆 3 孔”

第9步：在“模型”面板“传感器”模型列表中，选择“灰度传感器”添加到编辑区，参照图 7.11 将灰度传感器左下方第 2 排第 1 列的安装点，与“方杆 3 孔”组合左前方上数第 1 个安装点相连，再按 1 次空格键，使灰度传感器转 90 度，检测口向下。

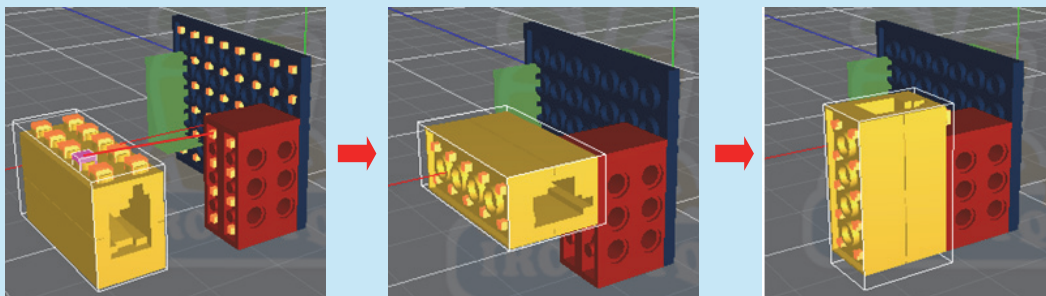


图 7.11 安装灰度传感器



第 10 步：将组合完成后的组件以“灰度传感器-1”为模板名，保存到“我的模板”中。

### 试试看

给“学习者 1 号”设计一个可以检测站台专用停车线——红色横线的“灰度传感器-2”模板组件。



## 2. 为“学习者 1 号”安装灰度传感器组件

完成了灰度传感器组件的搭建后，再给“学习者 1 号”安装“灰度传感器”就不再是难事儿，只要在 IRobotQ3D 的机器人编辑窗口打开“学习者 1 号”，将“灰度传感器-1”组件组装上去就可以了。



### 做一做

在“学习者 1 号”上安装“灰度传感器-1”组件，将其另存为“学习者 2 号”。

第 1 步：登录 IRobotQ3D，在仿真平台中打开 Robokid 机器人编辑窗口，在机器人编辑窗口中打开“学习者 1 号”。

第 2 步：在模型库“模板”列表中，单击“我的模板”文件夹中的“灰度传感器-1”，把它添加到编辑区“学习者 1 号”左侧，再单击“灰度传感器-1”中的“厚连杆 3×5 孔 L 型”，按 1 次空格键，使“灰度传感器-1”旋转 90 度，让灰度传感器向前，如图 7.12 所示。

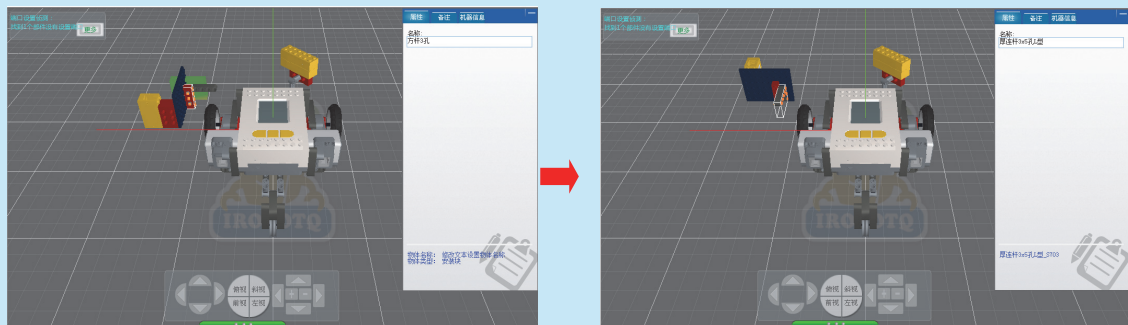


图 7.12 添加“灰度传感器-1”

第 3 步：在模型库“模型”面板“安装块”模型列表中，分别选择 2 次“转轴”，依次添加到编辑区“学习者 1 号”左侧，再单击“学习者 1 号”，按 1 次空格键，使其旋转 90 度，显示左侧大电机，如图 7.13 所示。

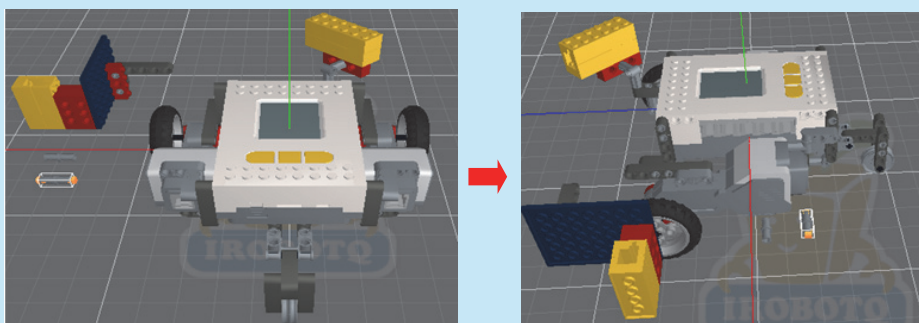


图 7.13 添加“转轴”并旋转控制器

第 4 步：参照图 7.14，将其中的 1 个“转轴”，安装到“学习者 1 号”左侧“厚连杆 7 孔”左数第 1 个安装孔内。

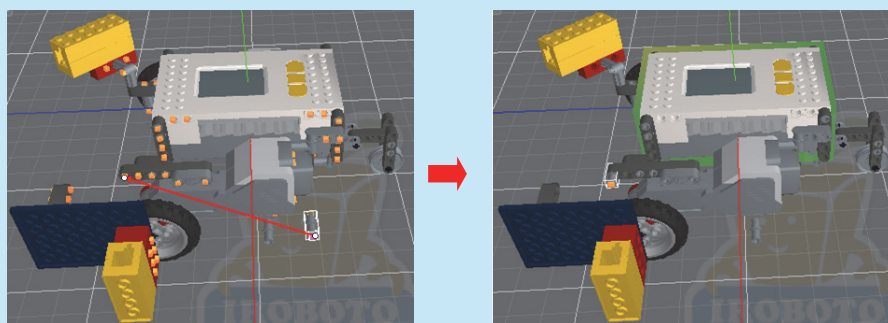


图 7.14 安装“转轴”到“厚连杆 7 孔”上

第 5 步：参照第 4 步和图 7.15，将另一个“转轴”安装到“学习者 1 号”左侧“厚连杆 7 孔”左数第 2 个安装孔内。

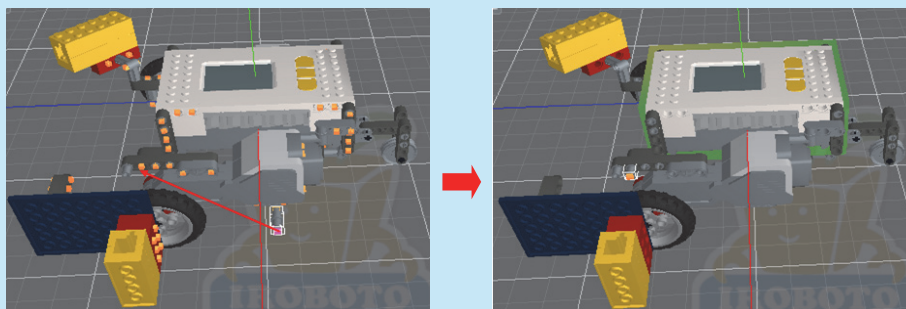


图 7.15 安装另一个“转轴”到“厚连杆 7 孔”上

第 6 步：单击“学习者 1 号”，按下鼠标右键向左拖动，使控制器正面朝前，参照图 7.16，将“灰度传感器-1”上“厚连杆 3×5 孔 L 型”最外侧的安装点，安装到“学习 1 号”左侧“厚连杆 7 孔”下数第 1 个“转轴”上，再按 2 次空格键，使“灰度传感器-1”旋转 180 度，让灰度传感器朝向正前方。

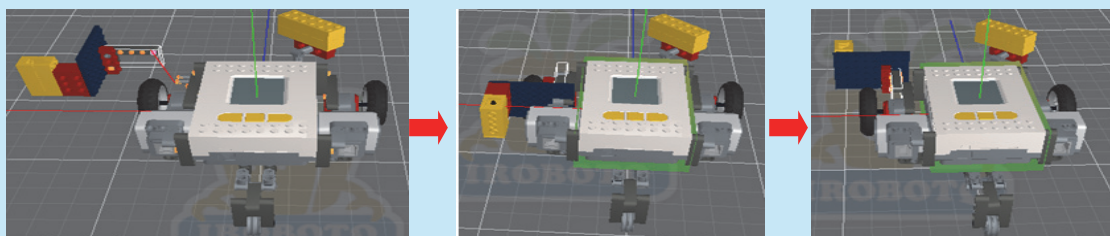


图 7.16 将“灰度传感器-1”安装到“学习者1号”上

**小博士**

由于要检测公路中间的双黄线，同时要“遵守交规”不能压到双黄线，所以要用“板6×8柱”将灰度传感器安装在机器人左前方且检测方向朝下。



第7步：在编辑区单击“学习者1号”左前方的灰度传感器，单击属性面板中的“属性”标签，在“设置端口”列表中选择端口32，如图7.17所示。

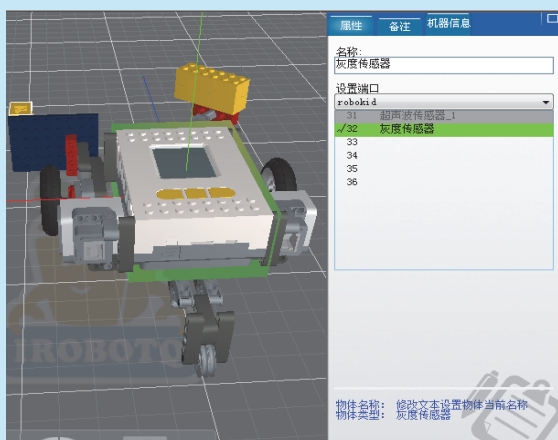



图 7.17 设置灰度传感器的属性

**小博士**

IRobotQ3D 中灰度传感器的最大检测距离为 5，返回值为 0~255，当物体色彩越趋向于白色时，其值越大，反之越小。可以使用小地图工具栏中的取色工具，到仿真窗口内进行“颜色取值”。



第8步：完成后以“学习者2号”为文件名保存机器人。

**说说看**

当给“学习者号”逐步安装传感器时，“学习者号”的重量也在增加，这会不会影响仿真效果？





### 试试看

要让“学习者 2 号”的灰度传感器检测公路中间的双黄线，且在行进过程中压不到双黄线，请进一步修改“学习者 2 号”灰度传感器的安装方式。



## 三、仿真体验使用灰度传感器

安装了灰度传感器后，“学习者 2 号”具备了识别道路上交通标识线的能力，我们还需要针对它的这一功能设计相应程序，控制“学习者 2 号”按照标识线规定，在道路上行驶。

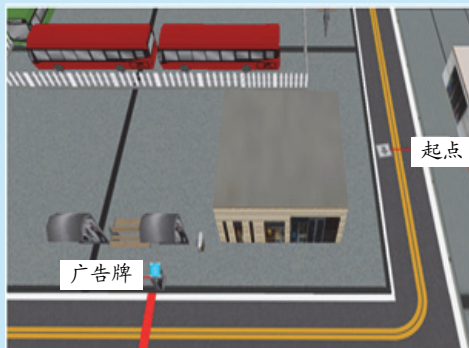


图 7.18 “遵守交规”仿真场地示意图



### 做一做

图 7.18 是 IRobotQ3D 提供的“遵守交规”仿真场地，让“学习者 2 号”从起点沿公路行驶到下一个公交站点接送乘客后，再继续前进。

#### 1. 任务分析

观察整个仿真场地，发现道路中央有一条明显的双黄线，让“学习者 2 号”在灰度传感器(检测方向向下)的控制下沿着双黄线的右侧一直前进，用超声波传感器检测站台广告牌实现停靠接客，就能保证在遵守交通规则的前提下完成任务。

#### 2. 算法描述

用流程图表示这个过程如图 7.19 所示。

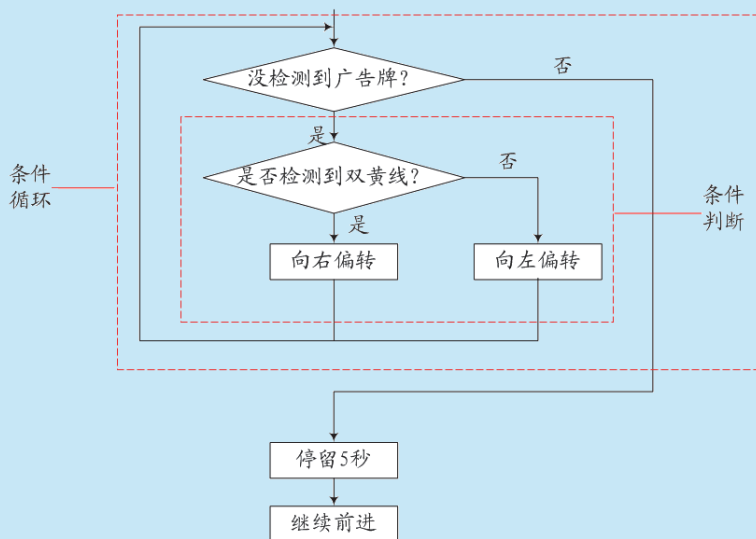




图 7.19 “遵守交规”流程图





### 3. 编写程序与仿真调试

第 1 步：登录 IRobotQ3D，在“仿真平台”中打开 LOGO 程序编写窗口。

第 2 步：单击“文件”标签工具栏中的 ，创建一个新的 LOGO 程序，再单击“文件”标签工具栏中的 ，以“logo\_遵守交规”为文件名保存程序，完成后关闭 LOGO 程序编写窗口，返回到仿真平台。

#### 小博士

当任务场景过于复杂，无法直接得知机器人前进或旋转等所需的具体参数时，可以新建一个空的程序文件，先进入仿真窗口，根据仿真场景获得关键的信息后，再编写具体的程序代码。






第 3 步：在仿真平台单击 ，进入快速仿真包窗口，单击控制区中的 ，按照界面上方的导航提示，依次在任务列表窗格中选择：“选择的任务”为场景\_遵守交规，“选择的机器人”为学习者2号，“选择的程序”为logo\_遵守交规，在 4. 导航完毕环节，在 保存仿真包：名称框中输入“遵守交规”，如图 7.20 所示，然后单击 保存进入 进入仿真窗口。



图 7.20 导航完毕界面

第 4 步：在仿真窗口，先单击场景小地图工具栏中的 ，再将鼠标指针移到仿真窗格的双黄线上单击，弹出“颜色取值”对话框，显示双黄线的灰度值为 161，如图 7.21 所示，单击对话框中的 确定，关闭对话框，用同样的方法获得双黄线一侧灰色地面的灰度值约为 83。



2. 单击双黄线 3. 弹出“颜色取值”对话框 1. 单击取色工具






图 7.21 取双黄线灰度值

### 小博士

1. 取灰度值作为判断条件时，通常取两个相临界面灰度值的平均值，例如在“遵守交规”仿真场景中，双黄线的灰度值是 161，而双黄线两侧路面的灰度值是 83 (实际测得双黄线两侧地面的灰度值分别是 75、83 和 90 值，这里取平均值 83)，它们的平均值约等于 120，所以可以用 120 作为判断双黄线的条件值。

2. 为了提高机器人仿真效率，通常只有与任务相关的指定物体的颜色可以被灰度传感器检测到，如轨迹、跑道、迷宫场地、颜色标记等，而大的地形、草地、山等物体，在灰度检测时总是返回 0。



第 5 步：单击仿真工具栏中的，打开 LOGO 程序编写窗口，在程序编辑区参照图 7.22 输入程序代码；再单击“文件”工具栏中的保存程序；最后单击 LOGO 程序编写窗口右上方的，返回到仿真窗口。

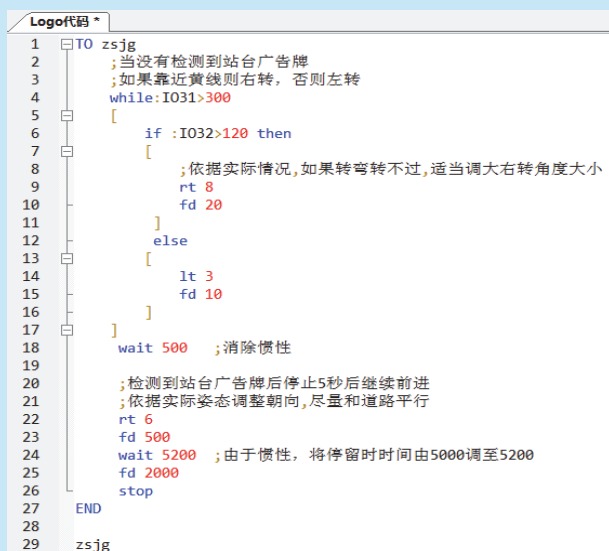


图 7.22 在程序编辑区输入程序代码




**小博士**

因为受到具体场景和机器人重量的影响以及兼顾后续任务的连续性，在程序调试过程中，往往会加入一些命令语句对机器人运行状态进行修正，比如在程序的末尾加入了一段程序：

```
rt 6
fd 500
wait 5200
fd 2000
stop
```

这样的情况在今后的仿真调试中会经常出现。



第6步：单击仿真工具栏中的, 开始仿真调试。

**知识窗****条件循环与条件判断的嵌套**

WHILE 条件循环较之尾递归更易于理解。与次数循环(REPEAT)相比，WHILE 循环的特点是，当无法明确循环的次数时，可以利用其自带的条件来控制循环是否继续执行。当 WHILE 条件循环与 IF 条件判断结合使用时，会让程序的功能更加强大，使机器人变得更加智能。

**1. “条件循环”(WHILE)命令**

条件循环在未知地形却有明确任务终点的“遵守交规”任务中，有着很大的用武之地，该命令的具体内容如表 7.2 所示。

表 7.2 “条件循环”命令

命 令	格 式	作 用
条件循环 (WHILE)	WHILE 条件 [ 循环体语句 ]	当满足某个条件时一直执行循环体语句，直到条件不满足之后结束循环

WHILE 循环的流程图如图 7.23 所示。

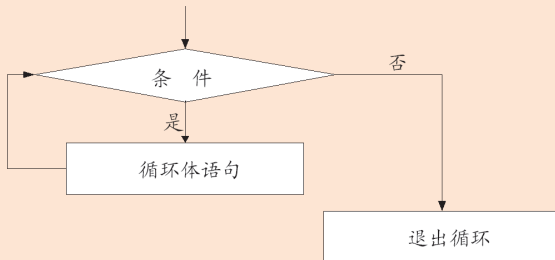


图 7.23 WHILE 条件循环流程图

在“遵守交规”任务中，WHILE 命令的作用是当没有检测到终点——站台广告牌时，“学习者 2 号”一直行驶，直到检测到站台广告牌时停止行驶。

## 2. 条件循环与条件判断的嵌套

条件判断不同于条件循环，条件循环在满足设置条件的情况下一直在循环体内执行，直到不满足设置条件时退出循环结构；而条件判断是个选择结构的程序，根据条件是否成立，执行某个分支程序并结束该模块结构。但条件判断结构所判断的条件与条件循环结构基本一致，如灰度的变化、光线的变化、海拔的变化、距离的变化等等，都可以作为其判断的条件。

在编写机器人程序时，条件判断常常会与条件循环结合使用，使得条件判断结构成为条件循环结构中的循环体，这样可以让条件判断不断地运行，这也被称为条件循环与条件判断的嵌套。



### 说说看

在“logo\_遵守交规”程序中，循环的条件是什么？条件判断中的“条件”又是什么？



### 试试看

让“学习者 2 号”准确地停靠在公交站台，既可以利用上一单元使用超声波传感器判断站台标识物的方法，也可以利用灰度传感器辨识站台专用停车线——红色横线来实现。尝试用“灰度传感器-2”替换“学习者 2 号”上的超声波传感器，完成“遵守交规”任务(提示：将修改后的机器人和程序均以别的文件名另存)。



## 四、学习实践

“学习者 2 号”不仅装备了超声波传感器，还装备了灰度传感器，在条件循环与条件判断嵌套程序的驱动下，在仿真环境中已经能“遵守交规”上路行驶了。现在我们尝试让“学习者号”在实验场地上“遵守交规”行驶。

### 1. 任务

- ① 给实物机器人“学习者号”添加灰度传感器。
- ② 将仿真实验中的“logo\_遵守交规”程序下载到“学习者号”上，并在真实





场地中进行调试、运行。

## 2. 组织实施

① 6~8 个同学组成 1 个小组，在机器人实验室操作。

② 操作步骤：

第 1 步：参照本单元“二、在 IRobotQ3D 中给机器人添加灰度传感器”中的“做一做”，给“学习者号”添加编号为“CG-HD-01”的灰度传感器。

第 2 步：将“logo\_遵守交规”程序从电脑上下载到“学习者号”上。

第 3 步：参考图 7.24，在光滑的平面上画出“遵守交规”场地。

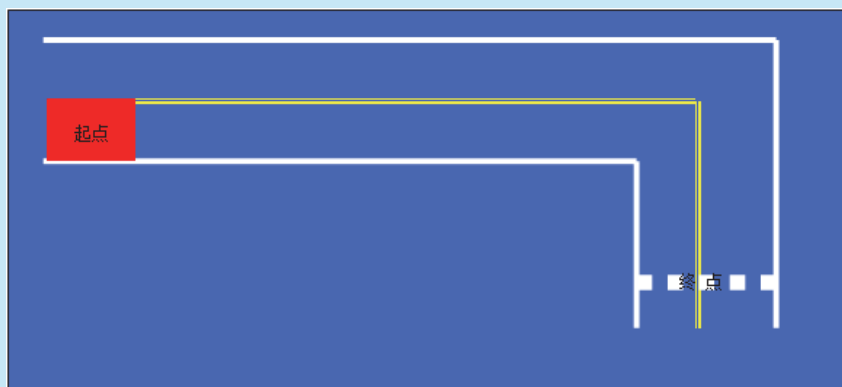
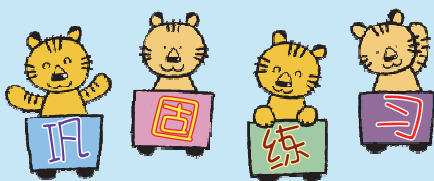


图 7.24 “遵守交规”场地示意图

第 4 步：在“遵守交规”场地上调试、运行“学习者号”。



1. 判断下列说法的正误。

① 灰度传感器利用不同颜色的检测面对光的反射程度不同的原理，进行颜色深浅检测。( )

② 当任务相当复杂时，我们往往先新建一个空的程序文件，进入仿真窗口后，根据场景获得关键的信息后，再编写具体的程序代码。( )

③ 在编写机器人程序时，条件判断常常与条件循环结合嵌套使用，使得条件判断结构成为条件循环结构中的循环体，让条件判断循环运行。( )

④ IRobotQ3D 中灰度传感器的最大检测距离为 5，返回值在 0~255 之间，纯白色的灰

度值是 0，黑色的灰度值是 255。( )

⑤ WHILE 条件循环与次数循环(REPEAT)可以通用，例如当明确知道循环的次数时。( )

2. 在 IRobotQ3D 中进行仿真时，对仿真测试结果影响最小的是( )。

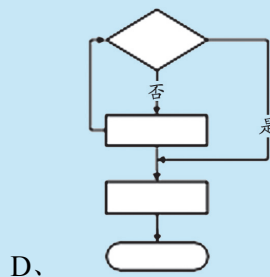
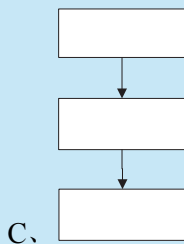
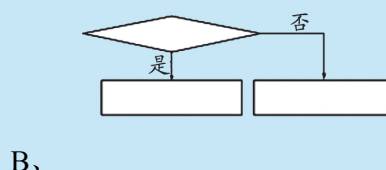
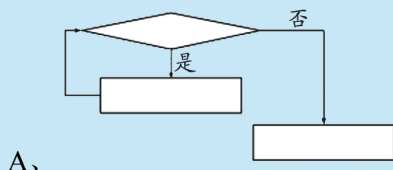
A、仿真场地

B、程序

C、机器人

D、安装 IRobotQ3D 软件的计算机

3. 下面的( )流程图是循环结构的流程图。



4. 如果在“遵守交规”仿真场景中，双黄线的灰度值是 161，双黄线两侧路面的灰度值是 83，在编写程序时，用下列哪个选项作为灰度值进行条件判断最合适( )。

A、161

B、83

C、120

D、255

5. 说一说，同样是实现循环，“WHILE 条件循环”与“尾递归”和“次数循环(REPEAT)”相比较，有什么不同？分别适合在什么情况下使用？

6. 在 IRobotQ3D 中如何获取灰度值？取灰度值作为判断条件时，有什么约定俗成的规定？

7. 利用公路右侧的白线，修改机器人和程序来完成“遵守交规”仿真任务。

8. 用条件循环与条件判断的嵌套，实现“精准停靠”仿真任务。

9. 用“尾递归”修改“logo\_遵守交规”程序，指挥“学习者 2 号”完成“遵守交规”仿真任务。





## 第8单元 展示技艺

“学习者2号”已经出色地完成了接迎乘客的任务，即将到达存放宝藏的迷宫(图8.1所示为“魔幻寻宝”的仿真场地)，接下来它将带着我们经历“魔幻寻宝”：首先寻找迷宫的入口，再穿越奇幻的迷宫，最后跨过三道封锁线拿到神秘宝物。显然，“魔幻寻宝”是一个系列任务，为了完成这个系列任务，还要为“学习者2号”增加能够检测障碍物的装备——“障碍传感器”和能够“拿到”宝物的装备——“触碰传感器”。

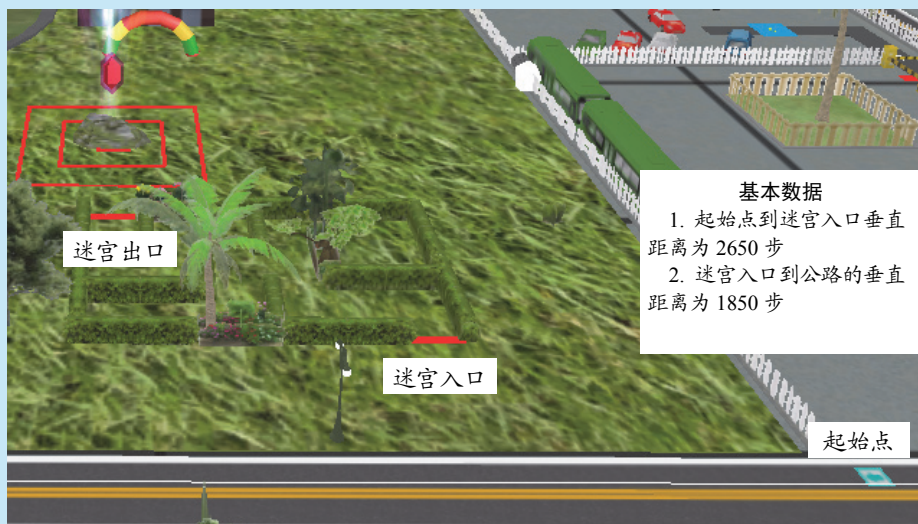


图 8.1 “魔幻寻宝”仿真场地示意图

### 一、障碍传感器和触碰传感器

障碍传感器和触碰传感器都是机器人常用的传感器，主要用于检测在一定的范围内有无物体，但两种传感器的工作原理不同。

#### 1. 障碍传感器的工作原理

障碍传感器是机器人常用的光敏传感器之一，从名称上就可以理解，它的功能是用来检测障碍物的。图8.2所示是“学习者号”使用的障碍传感器。



障碍传感器工作时会发射出一条蓝白相间的光线，在一定范围内，如果没有障碍物，发射出去的光线因为传播距离渐远而逐渐减弱，最后消失。如果有障碍物，光线遇到障碍物，会被反射到传感器接收头，传感器检测到这一信号，就可以确认有障碍物，并将信号传送给控制器，控制器调用程序指令，指挥机器人完成相应动作。



图 8.2 障碍传感器

障碍传感器的检测面在发光部件的一侧，安装时检测面应朝向要检测的物体。

障碍传感器的检测范围为 1~500，默认检测距离为 500，当障碍物与传感器之间的距离超过检测范围时，将检测不到障碍物。工作时，如果检测到障碍，返回结果用 1 表示；如果没有检测到障碍，返回结果用 0 表示。

## 2. 触碰传感器的工作原理

触碰传感器是一种机械传感器，用来检测是否发生与外界触碰的情况，图 8.3 所示是“学习者号”使用的触碰传感器。



图 8.3 触碰传感器

触碰传感器有个内置弹簧片和一个开关电路，当检测触点触碰物体受到较大的外力时，开关电路接通，表示检测到物体，否则，开关电路保持断开状态，表示一直没有触碰到物体。

触碰传感器的触碰面在有凸起弹簧的一侧，安装时触碰面要朝向触碰的物体。

触碰传感器一般安装在机器人最外缘的位置，使它能最先接触到触碰物体。工作时触碰到物体，返回结果用 1 表示；没有触碰到物体，返回结果用 0 表示。

## 二、在 IRobotQ3D 中给机器人添加障碍传感器和触碰传感器




在 IRobotQ3D 中给机器人添加障碍传感器和触碰传感器的方法，与添加其他传感器是一样的，都是在 Robokid 机器人编辑窗口“模型库”窗格的“传感器”列表中进行选择，所不同的是它们的性能参数不一样，安装时要注意传感器的方向，并根据需要设定其参数。




利用表 8.1 中的部件，在“学习者 2 号”上添加一个障碍传感器和一个触碰传感器，完成后将其另存为“学习者 3 号”。



表 8.1 安装障碍传感器和触碰传感器所需零部件清单

名 称	图 片	数 量	作 用
障碍传感器		1	检测墙壁
触碰传感器		1	触碰宝物
板 2×6 柱		2	用于障碍传感器与触碰传感器的连接

第 1 步：登录 IRobotQ3D，在仿真平台中打开 Robokid 机器人编辑窗口，在机器人编辑窗口中打开“学习者 2 号”。

第 2 步：在编辑区单击选中“学习者 2 号”，按 1 次空格键，使其旋转 90 度，安装正面朝左；连续单击视角面板中的，缩小显示比例至“学习者 2 号”完全显示在编辑区中，如图 8.4 所示。

第 3 步：在模型库“模型”面板的“传感器”模型列表中，选择“障碍传感器”添加到编辑区，按 5 次空格键，使其旋转到有凹槽安装点的一面向上，然后参照图 8.5，将障碍传感器右下方第 1 个安装点，安装到控制器左上方第 4 行第 1 列上。

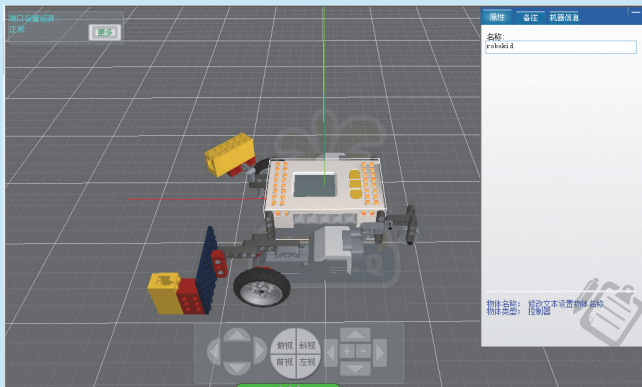


图 8.4 旋转、缩放“学习者 2 号”

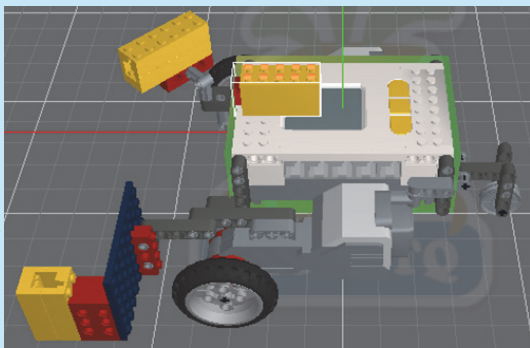
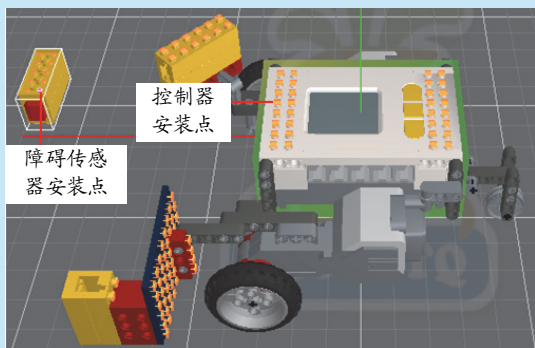


图 8.5 安装障碍传感器

第 4 步：在模型库“模型”面板的“安装块”模型列表中，分别选择 2 次“板 2×6 柱”，依次添加到编辑区，选中其中的一块“板 2×6 柱”，按 5 次空格键，使其旋转至底部凹槽向上，然后参照图 8.6，将“板 2×6 柱”左下角的安装点，安装到障碍传感器左下方第 1 行第 2 列的安装点上，再按 1 次空格键，旋转 90 度，将“板 2×6 柱”横向安装在障碍传感器正上方。



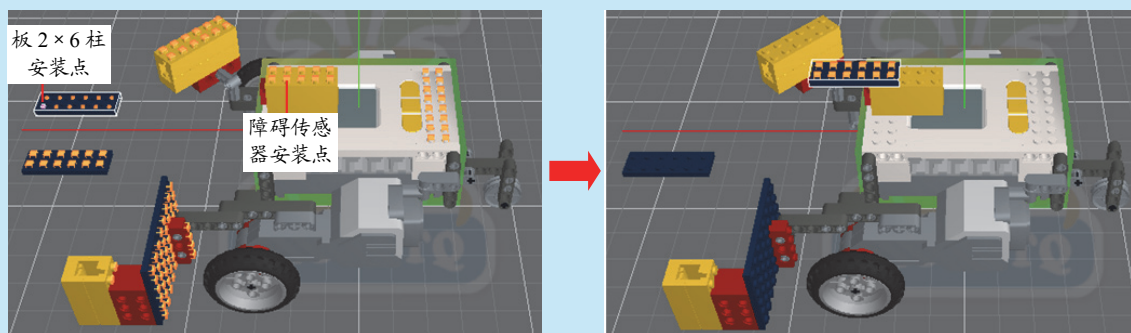


图 8.6 安装第 1 块“板 2×6 柱”

第 5 步：参照第 4 步和图 8.7，将另一块“板 2×6 柱”左上方的安装点，安装到第 1 块“板 2×6 柱”左上方第 1 行第 2 列安装点上。

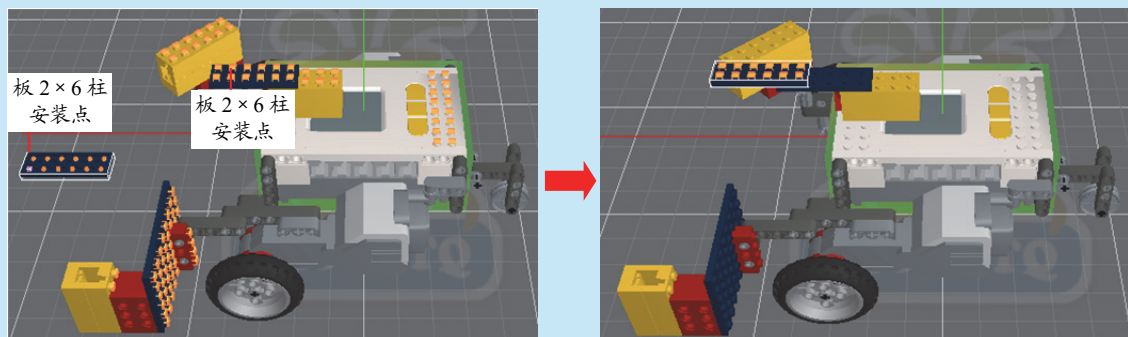


图 8.7 安装第 2 块“板 2×6 柱”

第 6 步：在模型库“模型”面板的“传感器”模型列表中，选择“触碰传感器”添加到编辑区，按 5 次空格键使其旋转到有凹槽安装点的一面向上，接着参照图 8.8，将触碰传感器左上角安装点，安装到左边“板 2×6 柱”左下方第 1 行第 3 列的安装点上，再按 3 次空格键，使触碰传感器旋转至检测面向左。

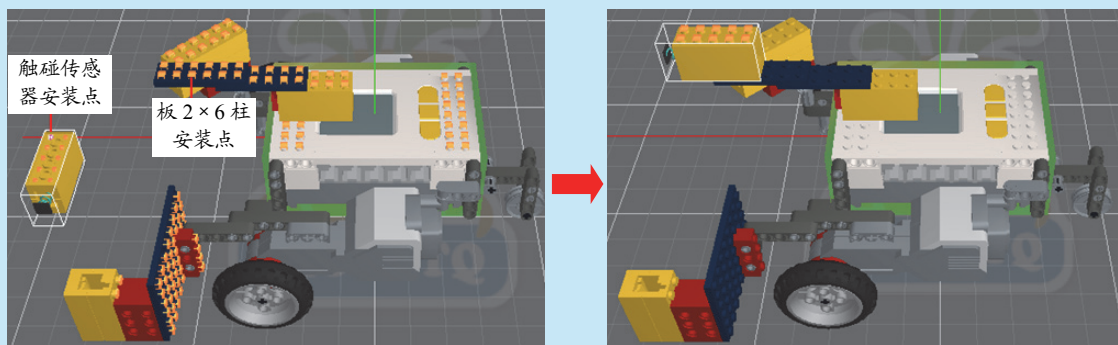


图 8.8 安装触碰传感器

第 7 步：在编辑区分别选择控制器上的障碍传感器和触碰传感器，参照图 8.9，在“属性”面板中分别设置其属性值为：



障碍传感器：“设置端口”设置为“33”，“检测距离”设置为“300”。

触碰传感器：“设置端口”设置为“34”。

第8步：以“学习者3号”为文件名保存机器人。

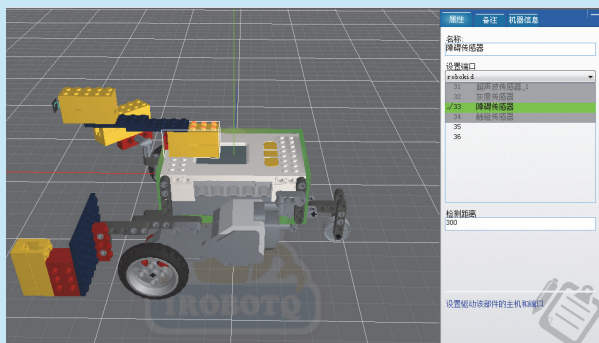


图 8.9 设置传感器的属性

### 说说看

为什么要如此安装触碰传感器？



### 试试看

将“学习者3号”右前方的超声波传感器换成障碍传感器，再另存为“学习者4号”。



## 三、仿真体验使用障碍传感器和触碰传感器

“学习者3号”添加了障碍传感器和触碰传感器后，“感觉器官”更加丰富，可以去完成闯迷宫寻宝的任务了。



### 做一做

参照图 8.1 描绘的“魔幻寻宝”仿真任务场景，编写程序，指挥“学习者3号”完成寻宝任务。

#### 1. 任务分析

如何才能走出迷宫获得宝物呢？仔细分析一下，可以将“魔幻寻宝”任务细化为三个小任务：寻找迷宫入口、走迷宫和取宝。其中：

① 寻找迷宫入口。通过仿真场地可以看到，“学习者3号”从出发点沿箭头所示方向“前进→右转→前进”，就可以找到迷宫入口。

② 走迷宫。需要解决的是，在没有找到迷宫出口前，要一直在迷宫中行走。因为迷宫出口处有红色标志线，可以利用灰度传感器来检测迷宫出口。

机器人走迷宫的策略：不管把它的起始点设置在哪里，只要它始终沿着某一侧墙壁走，肯定能走出迷宫。在这个过程中，要保证机器人与墙壁有一定距离，如果机器人太靠近墙壁，则要让它远离一些；如果机器人离墙壁太远，则要让它靠近一些。这种检测与左侧或右侧墙壁的距离，计算、制定机器人行走路径的程序设计计算



法，称为左手迷宫算法或右手迷宫算法。

左手迷宫算法：机器人检测它左边的墙壁，如果离左边墙壁太近，则机器人右转；如果离左边墙壁太远，则机器人左转；如果检测到前方墙壁，则机器人右转。

右手迷宫算法：机器人检测它右边的墙壁，如果离右边墙壁太近，则机器人左转；如果离右边墙壁太远，则机器人右转；如果检测到前方墙壁，则机器人左转。

在这个仿真任务中，我们采用右手迷宫算法，通过超声波传感器和障碍传感器沿着右侧墙壁走迷宫。

③ 取宝。在迷宫出口的正前方，宝物被三个交替工作的激光防护墙(大小不等的同心红色正方形)包围着，只要一直前行使触碰传感器碰到宝物，就能完成“取”宝任务。

## 2. 算法描述

这个复杂的任务可以由一个总流程图(图8.10)和三个分解的子流程图(图8.11、图8.12和图8.13)表示。

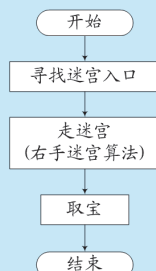


图 8.10 魔幻寻宝任务总流程图



图 8.11 寻找迷宫入口子任务流程图

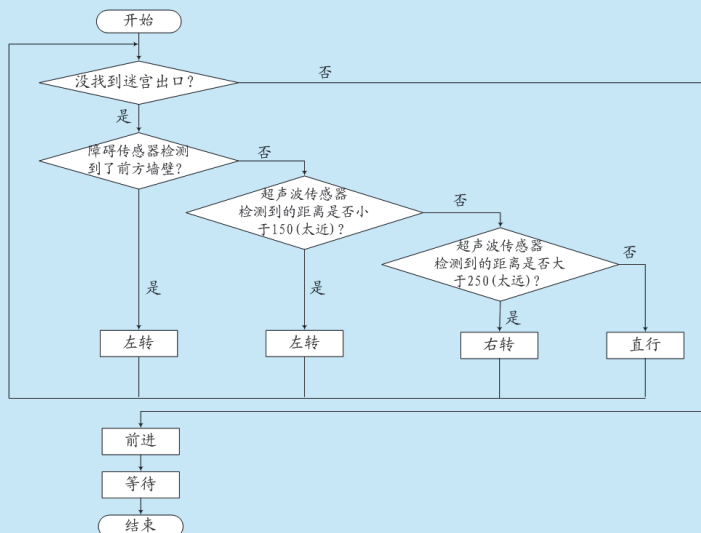


图8.12 走迷宫(右手迷宫算法)子任务流程图

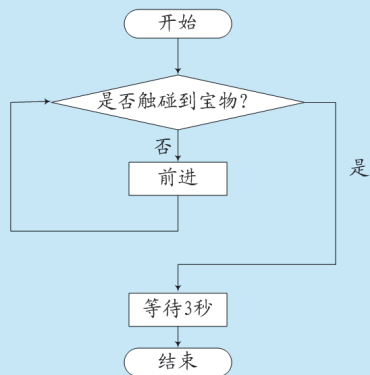


图8.13 取宝子任务流程图



## 3. 编写程序与仿真调试

第 1 步：登录 IRobotQ3D，在仿真平台打开 LOGO 程序编写窗口，创建一个新





的 LOGO 程序，以“logo\_魔幻寻宝”为文件名保存这个程序。完成后关闭 LOGO 程序编写窗口，返回到仿真平台。

第 2 步：在仿真平台单击，进入“快速仿真包”窗口，单击控制区中的，按照界面上方的导航提示，依次在任务列表窗格中选择：“选择的任务”为场景\_魔幻寻宝，“选择的机器人”为学习者3号、“选择的程序”为logo\_魔幻寻宝，在4. 导航完毕 环节的 保存仿真包：名称框中输入“魔幻寻宝”，如图 8.14 所示，然后单击 保存进入 进入仿真窗口。



第 3 步：在仿真窗口中，单击场景小地图工具栏中的，放大场景小地图，在地图中找到迷宫出口并单击，使仿真窗口中场景地图切换到迷宫出口周围，如图 8.15 所示，然后单击场景地图工具栏中的，恢复小地图。



图 8.14 导航完毕界面




图 8.15 放大的场景小地图

### 金钥匙

因为小地图显示的是按比例缩小的全部场景图，所以可以通过场景小地图在仿真窗格中显示某个看不到的场景位置。



第 4 步：单击场景小地图工具栏中的，分别在场景窗格中取迷宫出口红色标识线的灰度值和迷宫内路面的灰度值（本例分别为 76 和 0），如图 8.16 所示，所以迷宫出口灰度值的平均值约为 40。

第 5 步：单击仿真工具栏中

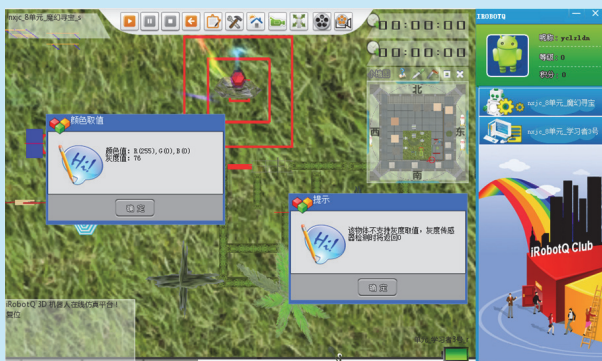



图 8.16 迷宫红色出口及其内部地面灰度值

的, 在 LOGO 程序编写窗口打开“logo\_魔幻寻宝”程序, 在程序编辑区输入图 8.17 和图 8.18 所示的程序代码。完成后保存程序, 关闭 LOGO 程序编辑窗口, 返回到仿真窗口。

```

,用右手迷宫算法走迷宫
TO mhxb2
while :IO32<40
[
,如果没有检测到迷宫出口
,如果前中障碍传感器检测到墙壁(前方有墙壁)则左转
,否则调用过程yq
if :IO33=1 then
[
lt 35
]
else
[
yq
]
]
fd 180
wait 3000
rt 6
]
END

```

图 8.17 “logo\_魔幻寻宝”程序代码 1

```

TO yq
,如果右侧超声波传感器检测到离墙壁太近则右转
,如果右侧超声波传感器检测到离墙壁太远则左转
,如果右侧超声波传感器检测到离墙壁不近不远则直行
if :IO31<150 then
[
lt 8
fd 20
]
else
[
If :IO31>250 then
[
rt 8
fd 25
]
else
[
fd 50
]
]
]
END

```

```


,利用触碰传感器取得宝物
TO mhxb3
While :IO34=0
[
fd 8
]
wait 3000
END

,过程调用
mhxb1
mhxb2
mhxb3

```

图 8.18 “logo\_魔幻寻宝”程序代码 2

### 金钥匙


在“走迷宫”的过程中, 由于不知道“学习者 3 号”与墙壁之间的距离多少才合适, 可以先写一个慢速往前走的程序, 开始仿真时, 在合适位置处单击仿真控制工具栏中的暂停按钮, 观察窗口左上角机器人信息区中超声波传感器和障碍传感器的返回值, 如图 8.19 所示, 然后利用这些值修改程序。

机器人信息区中显示各个设置了端口的部件的返回值



图 8.19 在仿真窗口观察传感器与墙壁距离的返回值



第6步：单击仿真工具栏中的, 开始仿真调试。

### 说说看

障碍传感器和超声波传感器在功能上有什么区别？在“魔幻寻宝”任务走迷宫过程中，能不能用障碍传感器代替超声波传感器？



### 知识窗

#### 程序的基本结构

程序可以分为三种基本结构，即顺序结构、选择结构和循环结构。掌握这三种基本结构，是学习程序设计的基本要求，是编写出结构清晰、易读易懂程序的前提。

##### 1. 顺序结构

在“logo\_魔幻寻宝”程序中，“mhxb1”（寻找迷宫入口）过程就是一个顺序结构的程序。运行该过程时，按照由上到下的顺序一行一行地执行代码。程序执行过程中没有分支、没有重复，这种程序结构称为顺序结构。顺序结构是一种最基本的程序结构。

##### 2. 选择结构

在“logo\_魔幻寻宝”程序的“yq”（检测右侧墙壁）过程中，当超声波传感器检测到离墙壁太近时，机器人就左转；当检测到离墙壁太远时，则右转。这种根据不同的条件判断来决定程序执行走向的结构称为选择结构。

##### 3. 循环结构

在“logo\_魔幻寻宝”程序的“mhxb3”（寻得宝物）过程中，“fd 8”是多次被执行的语句（没有触碰到宝物时要不断前进），这种重复执行某一段代码的程序，需要使用循环结构来实现。LOGO 语言中常用的循环语句是“次数循环(REPEAT)”、“尾递归”和“WHILE 条件循环”。

一个较大规模的程序，通常不会仅仅是一种结构，更多的是三种结构的混合，比如“logo\_魔幻寻宝”程序中的“mhxb2”过程。



### 试试看

观察仿真窗口的机器人信息区中超声波传感器的返回值，试着修改“logo\_魔幻迷宫”程序中障碍传感器和超声波传感器变量的参数值，以提高仿真效果（仿真任务完成后获得的分值）。



## 四、学习实践 —— “学习者号” 走迷宫

“学习者 3 号” 装备了一个超声波传感器、一个灰度传感器、一个障碍传感器和一个触碰传感器，根据迷宫算法，成功地完成了仿真环境的测试。现在我们可以尝试在实际场地中，用真实的机器人来完成“魔幻寻宝”任务了。

### 1. 任务

- ① 给实物机器人“学习者号”添加一个障碍传感器和一个触碰传感器。
- ② 将仿真实验中的“logo\_魔幻寻宝”程序，下载到“学习者号”上，并在真实场地中进行调试、运行。

### 2. 组织实施

- ① 6~8 个同学组成 1 个小组，在机器人实验室操作。
- ② 操作步骤：  
 第 1 步：参照本单元第二节的“做一做”，给实物机器人“学习者号”添加一个编号为“CG-ZA-01”的障碍传感器和一个编号为“CG-CP-01”的触碰传感器。  
 第 2 步：将“logo\_魔幻寻宝”程序从电脑下载到“学习者号”。  
 第 3 步：参考图 8.20，在光滑的平面上画出场地，其中迷宫用木板搭出。

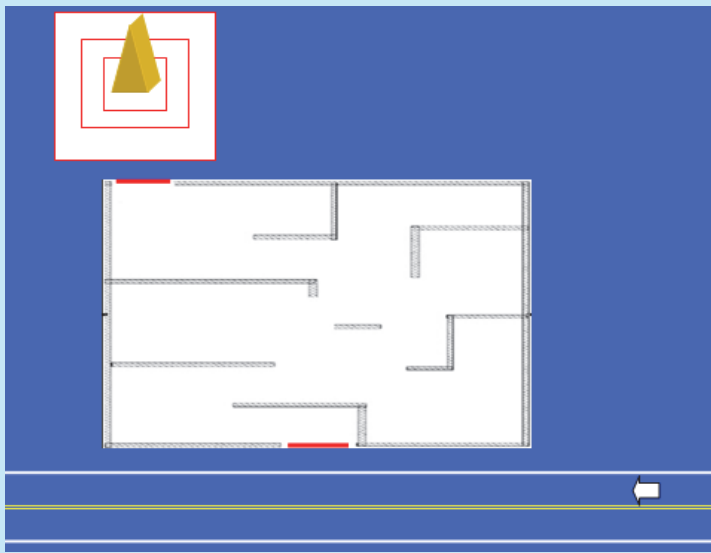
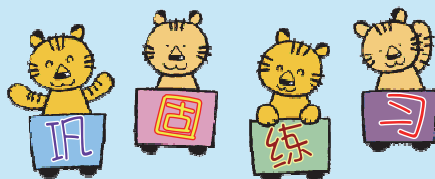


图 8.20 “魔幻寻宝” 场地示意图

- 第 4 步：在“魔幻寻宝”场地调试、运行“学习者号”。





1. 判断下列说法的正误。

- ① 宁宁说：“机器人用的传感器越多，执行任务就越快越精准”。( )
- ② 安装有触碰传感器的机器人在运行过程中，如果接触到物体，传感器会给控制器返回一个“1”，否则返回的是“0”。( )
- ③ 走迷宫是没有算法可言的，全凭运气！( )
- ④ 在 IRobotQ3D 中编写程序是离不开仿真场景的，往往要一边仿真测试，一边修改调试。( )
- ⑤ 小明说：“从原理上说，给‘学习者 3 号’安装两个超声波传感器，一样可以走出迷宫，但是因为在实际情况下两个超声波传感器在一起会相互干扰，因此安装两个超声波传感器的做法不合适。”( )

2. “logo\_安全出站”程序是( )结构的程序。

- A、顺序结构
- B、选择结构
- C、循环结构
- D、混合结构

3. “logo\_精准依靠”程序主要是以( )结构为主。

- A、顺序结构
- B、选择结构
- C、循环结构
- D、混合结构

4. 通常情况下，下列传感器中哪种传感器是机械传感器？( )

- A、灰度传感器
- B、超声波传感器
- C、触碰传感器
- D、障碍传感器

5. 请依据左手迷宫算法修改机器人和程序，指挥“学习者 3 号”完成“魔幻寻宝”任务(注：请将修改后的机器人和程序另存)。

6. 仔细观察“场景\_魔幻寻宝”，会发现宝物被三道红色封锁线包围着，试着利用这一条件修改机器人和“mhxb3”过程，指挥“学习者 3 号”完成“魔幻寻宝”任务(提示：应用“灰度传感器-2”模板和赋值命令(MAKE))。

7. 编写程序，指挥“学习者 4 号”完成“魔幻寻宝”任务。



“学习者号”相继(分别)完成了“安全出站”“精准停靠”“遵守交规”和“魔幻寻宝”一系列越来越复杂的任务。在这个过程中，我们体验了“学习者号”从一台“机器”逐步完善为“智能机器人”，现在，它能不能“一鼓作气”地在图 9.1 所示的场景中一次性全部完成上述任务呢？



图 9.1 “一鼓作气”任务场景

从机器人的搭建角度看，只要为“学习者号”配置合适的“感觉器官”，“一鼓作气”完成这些任务是没有问题的；但是从程序设计的角度来看，用一条条命令编写“一鼓作气”的程序会很长，调试难度较大。因此，“学习者号”能不能“一鼓作气”完成上述系列任务，需要解决的主要问题是程序设计的方法。

“学习者号”完成“魔幻寻宝”的过程给了我们启示：这个过程需要经历“寻找迷宫入口”“走迷宫”和“取宝”三个环节，我们将每个环节看做是独立的任务编写程序(参见上一单元给出的程序)，然后让“学习者 3 号”逐个调用程序，依次完成了三个任务，成功地“寻到宝藏”。

这种将一个大任务分解成若干个小任务，把每个小任务视作一个模块独立编写程序，再把若干小程序模块组成一个大任务程序的过程，称为模块化程序设计。

## 一、仿真体验模块化程序设计



让“学习者 3 号”在 IRobotQ3D 环境中，连续完成“安全出站”





“精准停靠”“遵守交规”和“魔幻寻宝”任务，简称“一鼓作气”，任务场景如图 9.1 所示。

### 1. 任务分析

“一鼓作气”要求“学习者号”一次就完成“安全出站”“精准停靠”“遵守交规”“魔幻寻宝”系列任务，需要为它同时装备“超声波传感器”“灰度传感器”“障碍传感器”和“触碰传感器”；然后将系列任务(过程)的每个环节看做是独立的小任务(子过程)编写程序，让“学习者号”依次调用。图 9.2 是基于上述分析设计“一鼓作气”程序结构图。

### 2. 算法描述

用流程图表示这个任务，如图 9.3 所示。

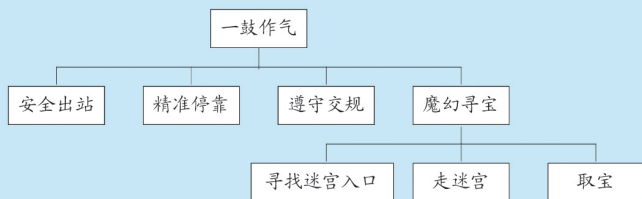


图 9.2 “一鼓作气”程序结构图

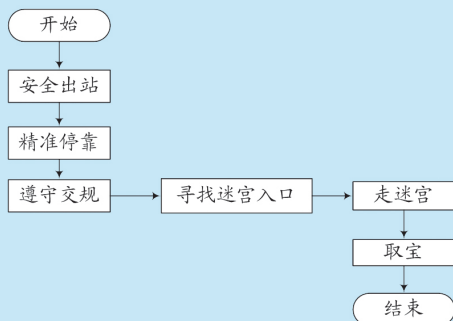

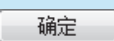



图 9.3 “一鼓作气”程序设计——流程图

### 3. 编写程序与仿真调试

第 1 步：登录 IRobotQ3D，由仿真大厅进入仿真平台，在 LOGO 程序编辑窗口中创建一个新的 LOGO 程序，然后以“logo\_一鼓作气”为文件名保存这个程序。

第 2 步：单击“文件”工具栏中的 ，在弹出的“打开文件”对话框文件列表中，选择“logo\_安全出站”文件，单击  按钮；在程序编辑区拖动鼠标指针，选定“aqcz”过程的全部代码，如图 9.4 所示；单击“操作”工具栏中的 。

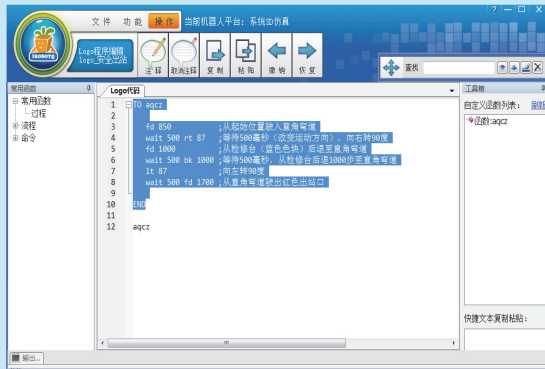
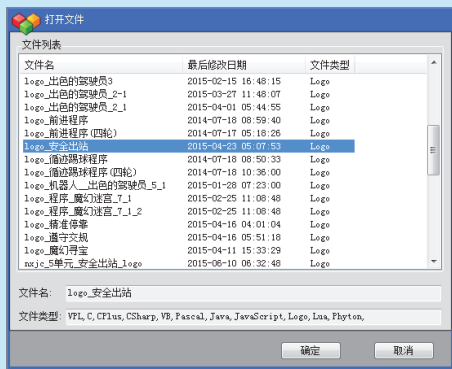


图 9.4 复制“aqcz”过程代码

# 金钥匙

1. 与其他 Windows 应用软件一样, 在 LOGO 程序编辑窗口中也可以单击右键, 在弹出的快捷菜单中选择命令(如图 9.5 所示), 还可以使用组合快捷键(如使用 **Ctrl** + **C**、**Ctrl** + **V** 进行复制、粘贴操作)。

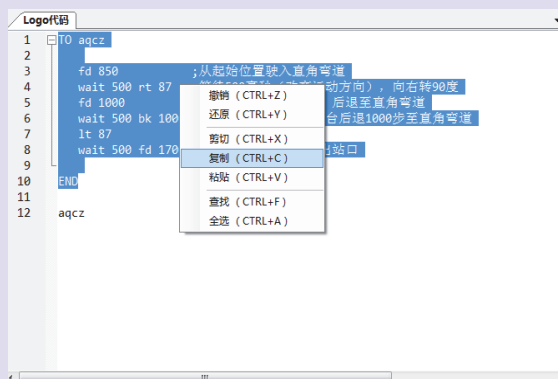


图 9.5 LOGO 程序编写窗口的快捷菜单

2. 每次复制的仅仅是各子程序中的过程, 而过程调用代码不需要复制。





第 3 步: 参照第 2 步, 在 LOGO 程序编辑窗口中打开“logo\_一鼓作气”, 将“aqcz”过程复制到程序编辑区(单击“功能”工具栏中的  或按 **Ctrl** + **V** 组合键), 如图 9.6 所示, 完成后单击“文件”工具栏中的 。




图 9.6 粘贴“aqcz”过程代码

第 4 步: 参照第 2 步和第 3 步, 依次将“logo\_精准停靠”“logo\_遵守交规”“魔幻寻宝”的过程, 复制到“logo\_一鼓作气”程序中。

第 5 步: 在完成复制的程序末尾, 参照图 9.7, 输入红线框内的“过程调用”程





序代码，完成后单击“文件”工具栏中的，然后关闭 LOGO 程序编辑窗口，返回仿真平台。

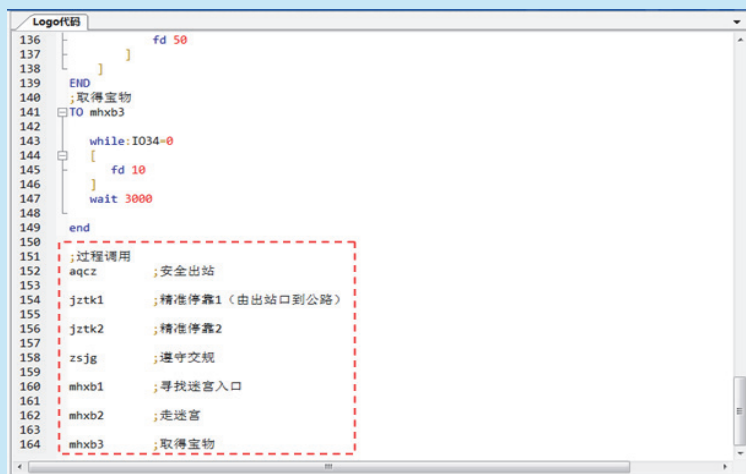


图 9.7 加入“调用过程”程序代码

### 金钥匙

为了方便过程的调用和程序的调试，在程序“logo\_一鼓作气”中，将原来的“logo\_精准停靠”过程修改为两个过程，具体内容如下：

;精准停靠

```
TO jztk1                ;机器人行至公路上以后右转弯
  fd 560
  wait 500
  rt 87                ;依据实际情况调整转向角度，与道路平行
  wait 500
END

TO jztk2                ;机器人寻找车站边上的广告牌
  fd 20                ;如果没发现站台上的广告牌则继续前进
  ;如果超声波传感器发现了站台边上的广告牌
  if :IO31<250 then
  [
    fd 400
    wait 5000
    fd 3000
  ]
  else
  [
    jztk2                ;尾递归调用
  ]
END
```








第 6 步：进入快速仿真包界面，单击控制区中的 ，按照界面上方的导航提示，依次在任务列表窗格中选择：“选择的任务”为场景\_一鼓作气，“选择的机器人”为学习者3号，“选择的程序”为logo\_一鼓作气，在4.导航完毕环节，在保存仿真包：名称框中输入“一鼓作气”，如图 9.8 所示，然后单击 ，进入仿真界面。



图 9.8 导航完毕界面

第 7 步：单击仿真工具栏中的 ，开始仿真。

### 知识窗

#### 程序的测试和微调

在多个子过程构成的程序中，每个子任务的起始位置对机器人的所处位置和行进角度要求较高，所以需要程序进行微调。

另外，为了提高仿真的可视效果，还可以在程序中加入“PRINT”打印命令，具体内容见表 9.1。

表 9.1 “打印”命令

命 令	格 式	作 用
打印(PRINT)	PRINT [字符串]	在界面上显示字符串内容

以“logo\_一鼓作气”程序为例，由于“一鼓作气”整个任务是多个小任务连接在一起而组成的，完成前一个任务后的结束状态不一定符合下一个任务起始状态的要求，此外还要顾及“学习者号”不断增加的重量(每完成一个任务都会增加不同的传感器)，以及仿真时同一程序在不同性能的计算机消耗 CPU 资源不同造成的仿真效果不同，所以就不能单纯地将前面的各个子程序简单地连接在一起，还要考虑每个任务之间如何“衔





接”。这就需要在程序“logo\_一鼓作气”的各个过程中,加入一些程序代码或修改一些参数,对程序进行必要的仿真测试和微调。

下面给出各个过程微调后的代码,其中绿色框中是修改后的参数,红色框中是添加的程序代码。

;安全出站

TO aqcz

print [任务一开始]

wait 500

fd 800

;从起始位置驶入直角弯道

wait 500 rt 87

;等待 500 毫秒(改变运动方向),向右转 90 度

fd 1000

;从直角弯道前进至检修台(蓝色色块)

wait 500 bk 1000

;等待 500 毫秒,从检修台后退 1000 步至直角弯道

lt 86

;向左转 90 度

wait 500 fd 1700

;从直角弯道驶出红色出站口

lt 87

;左转 90 度,与任务二起始点方向保存一致

END

;精准停靠

TO jztk1

;机器人行至公路上以后右转弯

print [任务二开始]

wait 500

fd 550

wait 500

rt 86

;依据实际情况调整转向角度,与道路平行

wait 500

END

TO jztk2

;机器人寻找停车站牌

fd 50

;如果没发现站台边上的广告牌则继续前进

;如果超声波传感器发现站台边上的广告牌

if :IO31 < 300 then

[

fd 400

print [已经到达站台]

wait 5200

;由于惯性,将停留时间由 5000 调至 5200

fd 1800

wait 500

]

else

[

jztk2

;尾递归调用

]

END

```

;遵守交规
TO zsjg
;当没有检测到站台广告牌时，如果靠近黄线则右转，否则左转
while :IO31>300
[
    if :IO32>120 then
    [
        rt 10 ;本单元机器人比第 7 单元的重，需要适当调整转向角度
        fd 20
    ]
    else
    [
        lt 4
        fd 10
    ]
]
wait 500 ;消除惯性
;依据实际情况调整转向角度，与道路平行
;检测到站台广告牌，停止 5 秒后继续前进
rt 2
fd 500
wait 5200
fd 2000
END

;魔幻寻宝
;寻找迷宫入口
TO mhxb1
print [任务四开始]
wait 500
fd 3000
wait 500
rt 85
wait 500
fd 1850
wait 2000
END

;右手迷宫算法
TO mhxb2
;当没有检测到迷宫出口(灰度传感器没有检测到出口地面红线)时
while :IO32<40
[
    if :IO33=1 then

```





```
[
    lt 35 ;如果前中障碍传感器检测到墙壁(前方有墙壁)则左转
]
else
[
    yq ;否则调用过程 yq
]
]
fd 180
wait 3000
lt 1 ;依据实际情况调整转向角度,以对准宝物
END
;检测右侧墙壁
TO yq
;如果右侧超声波传感器检测到离墙壁太近则左转,避免撞右侧墙
if :IO31<150 then
[
    lt 8
    fd 20
]
else
[
;如果右侧超声波传感器检测到离墙壁太远则右转,
;避免撞左侧墙或墙壁转弯处
if :IO31>250 then
[
    rt 8
    fd 25
]
else
[
    fd 50 ;如果右侧超声波传感器检测到离墙壁不近不远,则直行
]
]
]
END
;取得宝物
TO mhxb3
while :IO34=0
[
    fd 10
]
wait 3000
END
```

```

;过程调用
aqcz          ;安全出站
jztk1         ;精准停靠 1(由出站口到公路)
jztk2         ;精准停靠 2
zsjg          ;遵守交规
mhxb1         ;寻找迷宫入口
mhxb2         ;走迷宫
mhxb3         ;取得宝物

```



### 说说看

在完成整个“一鼓作气”仿真任务的过程中，直接影响仿真成功的因素有哪些？



### 试试看

请进一步修改和调试“logo\_一鼓作气”程序，提高仿真的稳定性和仿真分值。



## 二、学习实践

在上一单元“学习者号”就“全身披挂”上阵了：装备了一个超声波传感器、一个灰度传感器、一个障碍传感器和一个触碰传感器，现在让它在“logo\_一鼓作气”程序的指挥下，在实验场地中完成从出站到寻到宝藏的全部任务。

### 1. 任务

将仿真实验中的“logo\_一鼓作气”程序下载到“学习者号”机器人上，并在真实场地中进行调试、运行。

### 2. 组织实施

① 6~8 个同学组成 1 个小组，在机器人实验室操作。

② 操作步骤：

第 1 步：将“logo\_一鼓作气”程序从电脑上下载到“学习者号”上。

第 2 步：参照“安全出站”、“精准停靠”、“遵守交规”及“魔幻寻宝”四个场地制作“一鼓作气”场地。

第 3 步：在“一鼓作气”场地对“学习者号”中的传感器、程序进行调试、运行。

第 4 步：各小组互相比试(填写表 9.2 统计最终成绩)。



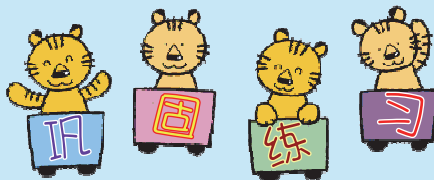




表 9.2 “一鼓作气”任务比赛小组成绩统计表

小组名称	比赛轮次	所用时间	本轮名次	成功次数	机器人外观(10分)	稳定性(20分)	总成绩
第1组	1						
	2						
	3						
第2组	1						
	2						
	3						
.....							

积分规则：比赛共进行3轮，每轮成功的小组按照所用时间进行排序，第一名加10分，第二名加8分，第三名加6分……没有成功的计0分。



1. 判断下列说法的正误。





① 将一个大任务分解成若干个小任务，把每个小任务视作一个模块独立编写程序，再把若干个小的程序模块组成一个大任务程序的过程，称为模块化程序设计。（ ）

② 使用模块化程序设计的思想，可以将一个复杂的问题分解成一个个简单的小问题，降低编程的难度。（ ）

③ 从总体上讲，“logo\_一鼓作气”是一个顺序结构的程序。（ ）

④ “PRINT”打印命令的功能是将所有程序代码都打印出来。（ ）

⑤ 由于“一鼓作气”仿真任务是由多个仿真任务组合在一起的，完成每个任务后的结束状态未必符合下一个任务起始状态的要求，所以要在各个过程中加入一些程序代码或修改一些参数，对程序进行必要的微调和仿真测试。（ ）

2. 下列哪种方法是 LOGO 程序编辑窗口不支持的复制/移动方法。( )
- A、单击右键在弹出快捷菜单中选择命令
  - B、按组合快捷键(如  + 、 + )
  - C、单击“操作”工具栏中的快捷图标
  - D、单击“常用”工具栏中的快捷图标
3. 如果任务复杂，程序代码过长，我们常常要想办法提高程序的可读性，以方便调试程序，下列哪些方法不可取。( )
- A、在合适的位置处加入备注语句
  - B、给程序中每个过程起一个“望文生义”的过程名
  - C、在合适的位置加入“PRINT”命令
  - D、用心记住每一段程序的功能与作用
4. 对“logo\_一鼓作气”程序进行微调和仿真测试时，最实用的方法是( )。
- A、改变机器人的运行角度
  - B、将所有程序推翻，重新编写
  - C、加入“wait”命令消除机器人的运动惯性
  - D、对程序一段一段地进行调试，不参与调试的部分可以用“;”进行注释
5. 什么是模块化的程序设计思想？说一说你的理解。
6. 在“logo\_一鼓作气”程序的合适位置添加“PRINT”命令，以提高程序的仿真可视性。
7. 对于相同的任务，在仿真环境下完成与在真实环境下完成有什么异同点？





## 第10单元 从 LOGO 语言编程到 可视化语言编程




在程序设计中, 将一个大的、复杂的程序按功能分解成为一些功能单一、结构清晰、容易理解的功能模块, 再为这些小的功能模块编写代码, 可以大大简化编写程序的难度。通常, 我们将常用的任务编写成一个个小程序模块, 把每一个小程序模块作为一个部件(如控件)保存下来, 形成程序模块库, 编写具体任务的程序时, 从库中调用需要的部件即可。这个编程思路可以转化为另一种编程方法——可视化程序设计。

### 一、可视化程序设计

VPL 是 Visual Programming Language 的缩写, 意为“可视化编程语言”。可视化程序设计是一种全新的程序设计方法, 是指程序设计人员利用软件本身所提供的各种控件, 像搭积木式地构造应用程序的各种界面。可视化程序设计最大的优点是设计人员可以不用编写或只需编写较少的程序代码, 就能完成应用程序的设计, 这样就能极大地提高设计人员的工作效率。

能进行可视化程序设计的计算机语言环境很多, IRobotQ3D 仿真平台中的“VPL”窗口就是一个可视化程序设计平台。对于在 IRobotQ3D 中搭建的机器人, 它既可以听从用 LOGO 语言编写的代码程序命令, 又可以听从用 VPL 编写的可视化程序命令。

### 二、熟悉 IRobotQ3D 的 VPL 编程环境

启动 IRobotQ3D 后, 在仿真大厅或仿真平台上单击“编写程序”功能按钮, 打开“选择编程平台”界面, 选择  就能打开“VPL 程序编辑”窗口初始界面, 如图 10.1 所示。

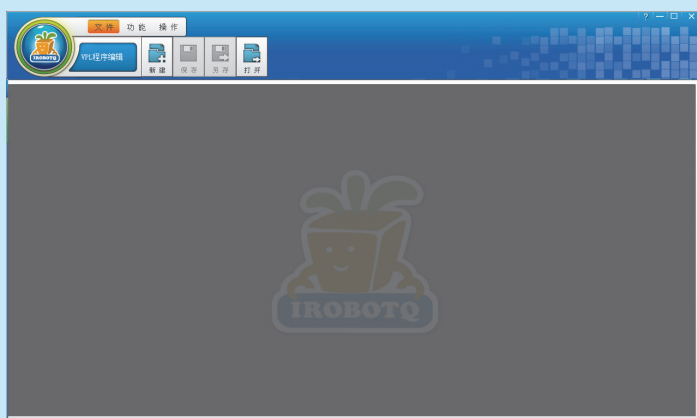





图 10.1 “VPL 程序编辑”窗口初始界面

在 IRobotQ3D 的“VPL 程序编辑”窗口中编程,需要先选择一个关联的机器人,使 VPL 程序获取机器人硬件的相关信息,如直流电机和传感器的数量、名称、端口设置等,才能进入编程界面。



熟悉 IRobotQ3D 的 VPL 程序编辑环境。

第 1 步:启动 IRobotQ3D,在仿真平台中单击打开“选择编程平台”界面,选择后单击,打开“VPL 程序编辑”窗口。



第 2 步:在“VPL 程序编辑”窗口“文件”工具栏中单击,在打开的“选择机器人”界面左侧列表窗格中选择机器人(这里以“学习者号”为例),如图 10.2 所示;单击,进入“VPL 程序编辑”窗口,如图 10.3 所示。



图 10.2 “选择机器人”界面



图 10.3 “VPL 程序编辑”窗口

### 小博士

“VPL 程序编辑”窗口与“机器人编辑”窗口相仿，但模型库中的内容不一样。

“VPL 程序编辑”窗口模型库中的每个模型都分别对应着程序中的一段代码，我们不用考虑“模型”是如何设计的，只要直接拿来用就可以了。



第3步：在模型库中选择不同的模型后，在 VPL 程序编辑区单击，把它添加到程序编辑区中，然后在属性面板中查看模型类型和内容。

### 说说看

启动“VPL 程序编辑”窗口与启动 **ROBO-LOGO** 编程窗口有什么不同？



### 试试看

指出“VPL 程序编辑”窗口与“机器人编辑”窗口的相同处和不同处。








### 三、体验 VPL 编程




在 IRobotQ3D “VPL 程序编辑”窗口编写 VPL 程序，只要根据任务需求，在模型库中选择合适的模型放置到程序编辑区，再通过属性面板对每个模型进行属性设置——如给变量赋值等，最后照一定的顺序将这些模型连接在一起，就能组合成结构直观、功能明确的 VPL 程序。



#### 做一做

为第 5 单元中的“安全出站”任务编写 VPL 程序，以“vpl\_安全出站”为程序名保存，并进行仿真调试。



第 1 步：登录 IRobotQ3D，在仿真平台单击，打开“选择编程平台”界面，选择后单击，打开“VPL 程序编辑”窗口。


第 2 步：在“文件”工具栏中单击，打开“选择机器人”界面，在机器人列表窗格选择“学习者号”，单击，返回“VPL 程序编辑”窗口，程序编辑区中出现绿色“开始”模块，参见图 10.3。

#### 小博士

新创建的 VPL 程序会自动创建一个“开始”模块，每个 VPL 程序中必须且只能有一个“开始”模块，它是 VPL 程序的开始标志，也是执行 VPL 程序时的入口。



第 3 步：在模型库的“模型”面板“驱动”列表中，单击后将鼠标指针移到编辑区下方再单击，将“多电机驱动”模块放入编辑区，如图 10.4 所示。

第 4 步：单击编辑区的，在右侧面板中单击“属性”标签，在属性面板中参照图 10.5 设置各项参数。

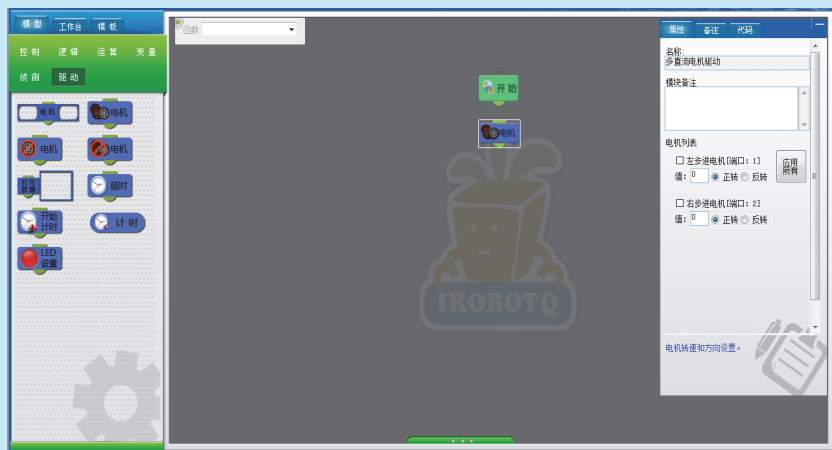


图 10.4 添加“多电机驱动”模块

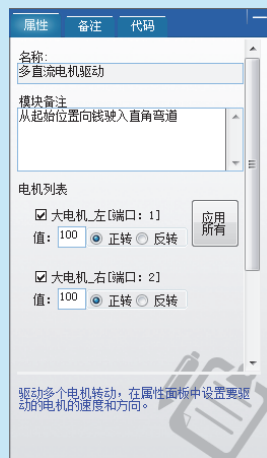


图 10.5 设置“多电机驱动”模块的属性



**小博士**

在 LOGO 环境中没有专门的电机驱动命令，所以机器人都以统一的速度行驶。而在 VPL 编程环境中，可以给各个电机设置不同的转速，灵活性较强。要让机器人前进，就给所有电机设置正向转速值；要让机器人后退，就给所有电机设置反向转速值；要让机器人左转，就让左侧电机反转，右侧电机正转；要让机器人右转，就让右侧电机反转，左侧电机正转。







第 5 步：将鼠标指针移到  下方的绿色半圆出口连接点处(出口连接点立即放大为黄红相间的半圆)。按下鼠标左键向下拖动到  上方方形入口连接点处松开左键， 与  之间出现一条闪动的连接线，如图 10.6 所示。



图 10.6 连接“开始”模块与“多电机驱动”模块

**小博士**

模块的连接点有方形和圆形两种，方形的是入口连接点，圆形的是出口连接点，如图 10.7 所示。编程时，由模块的圆形“出口连接点”去连接另一模块方形“入口连接点”。

方形入口连接点



圆形出口连接点

被选中的圆形出口连接点

图 10.7 模块连接点的形状



第 6 步：参照第 3 到第 5 步，依次添加表 10.1 所示的各模块，设置其属性并进行连接。

表 10.1 各模块的属性设置

顺序	模块名称	图 标	属 性	设 置 值
2	延时		名称	延时
			模块备注	直行延时
			延时(毫秒)	2000
3	多电机驱动		名称	多直流电机驱动_2
			模块备注	右转 90 度

顺序	模块名称	图 标	属 性	设 置 值
			大电机_左(端口: 1)	勾选
			值	30
			正转	选中
			大电机_右(端口: 2)	勾选
			值	30
			反转	选中
4	延时		名称	延时_1
			模块备注	右转延时
			延时(毫秒)	1220
5	停止所有步进与直流电机		名称	停止所有步进与直流电机
			模块备注	消除转弯惯性
6	延时		名称	延时_2
			模块备注	停止电机延时
			延时(毫秒)	500
7	多电机驱动		名称	多直流电机驱动_3
			模块备注	向前到检修台
			大电机_左(端口: 1)	勾选
			值	100
			正转	选中
			大电机_右(端口: 2)	勾选
			值	100
8	延时		名称	延时_3
			模块备注	向前延时
			延时(毫秒)	2600
9	停止所有步进与直流电机		名称	停止所有步进与直流电机_2
			模块备注	消除前进惯性
10	延时		名称	延时_4
			模块备注	停止电机延时
			延时(毫秒)	500
11	多电机驱动		名称	多直流电机驱动_4
			模块备注	后退至直角弯道
			大电机_左(端口: 1)	勾选
			值	100
			反转	选中
			大电机_右(端口: 2)	勾选
			值	100
12	延时		名称	延时_5
			模块备注	后退延时



顺序	模块名称	图 标	属 性	设 置 值
13	多电机驱动		延时(毫秒)	2500
			名称	多直流电机驱动_5
			模块备注	左转 90 度
			大电机_左(端口: 1)	勾选
			值	30
			反转	选中
			大电机_右(端口: 2)	勾选
			值	30
14	延时		正转	选中
			名称	延时_6
			模块备注	左转延时
15	停止所有步进与直流电机		延时(毫秒)	1550
			名称	停止所有步进与直流电机_3
			模块备注	消除转弯惯性
16	延时		名称	延时_7
			模块备注	停止电机延时
			延时(毫秒)	500
17	多电机驱动		名称	多直流电机驱动_6
			模块备注	驶出红色出站口
			大电机_左(端口: 1)	勾选
			值	100
			正转	选中
			大电机_右(端口: 2)	勾选
			值	100
			正转	选中
18	延时		名称	延时_8
			模块备注	向前延时
			延时(毫秒)	4500

### 金钥匙

当 VPL 程序太长、继续添加模块有困难时,有 3 种方法可以解决:



1. 单击“操作”工具栏(如图 10.8 所示)中的  , VPL 程序中各模块会立即排列得整齐、有序而紧凑。



图 10.8 “操作”工具栏

2. 将鼠标指针移到 VPL 程序任意一侧,然后按住鼠标左键向上或向下拖动。



3. 单击“操作”工具栏中的 ，进入全屏幕编写窗口，如图 10.9 所示。

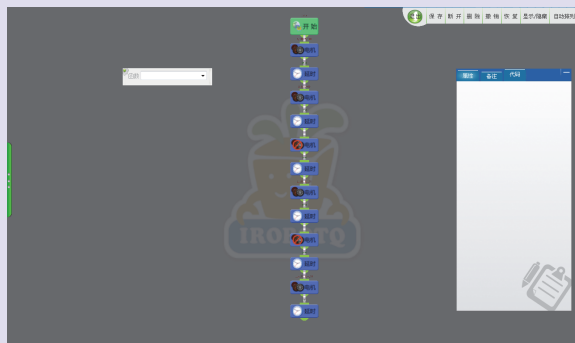



图 10.9 全屏幕编写窗口

第 7 步：单击“文件”工具栏中的 ，打开“保存控制程序”界面，以“vpl\_安全出站”为文件名保存 VPL 程序，如图 10.10 所示，然后关闭“VPL 程序编辑”窗口，返回仿真平台。



第 8 步：进入快速仿真包界面，单击控制区中的 ，按照界面上方的导航提示，依次在任务列表窗格中选择：“选择的任务”为场景\_安全出站，“选择的机器人”为学习者号、“选择的程序”为 vpl\_安全出站，在 4. 导航完毕 环节，在 保存仿真包 名称框中输入“安全出站\_vpl”，如图 10.11 所示，然后单击 保存进入 进入仿真界面。



图 10.10 保存 VPL 程序



图 10.11 导航完毕界面

第 9 步：单击仿真工具栏中的 ，开始仿真。

### 说说看

“vpl\_安全出站”程序中有许多重复的模块，利用已有的知识，你可以采用哪些方法提高编写程序的效率。





**试试看**

试着修改“vpl\_安全出站”程序，提高仿真分值。

**四、学习实践**

将用 IRobotQ3D 编写并测试成功的 VPL 程序下载到“学习者号”上，让它在“vpl\_安全出站”程序指挥下，再次完成“安全出站”任务。

**1. 任务**

将“vpl\_安全出站”程序下载到“学习者号”上，并在真实场地中进行调试、运行。

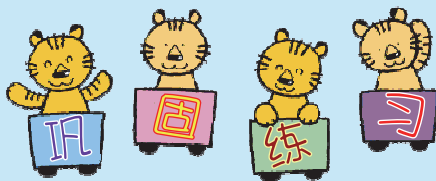
**2. 组织实施**

① 6~8 个同学组成 1 个小组，在机器人实验室操作。

② 操作步骤：

第 1 步：将“vpl\_安全出站”程序从电脑下载到“学习者号”上。

第 2 步：在“安全出站”场地调试、运行“学习者号”。

**1. 判断下列说法的正误。**

① 可视化程序设计是指程序设计人员利用软件本身所提供的各种图形化模块，像搭积木式地构造应用程序界面或构造应用程序。（ ）

② 在 Robokid 机器人编辑窗口构建的机器人，只支持 logo 程序设计语言，不支持 VPL 程序设计语言。（ ）

③ 在“VPL 程序编辑”窗口初始界面，不能直接新建一个 VPL 程序，必须先选择一个相关联的机器人，才能进入编程界面。（ ）



④ “VPL 程序编辑”窗口与“机器人编辑”窗口一模一样，没有什么区别。（ ）

⑤ 为了增强 VPL 程序的可读性，最常用的方法是设置关键模块的“模块备注”属性。（ ）

2. 编写 VPL 程序指挥机器人转弯，下列方法中，（ ）无法实现这一要求？

- A、给左、右电机设置不一样的转速
- B、将左、右电机中某一侧设置为“反转”属性
- C、将左、右电机中某一侧转速设置为 0
- D、将左、右电机的转速同时设置为 0

3. 在编写 VPL 程序时, 如果程序太长, 继续添加模块有困难, 下列方法中, ( ) 不能解决这一问题。

- A、单击“操作”工具栏中的 , 进入全屏编写窗口
- B、将鼠标指针移到 VPL 程序任意一侧, 然后按下鼠标右键拖动
- C、将鼠标指针移到 VPL 程序任意一侧, 然后按下鼠标左键拖动
- D、单击“操作”工具栏中的  自动排列

4. 在编写 VPL 程序时, 为了消除机器人的惯性, 通常采取( )方法?

- A、加入“停止所有步进与直流电机”模块
- B、加入“延时”模块
- C、加入“停止所有步进与直流电机”模块和“延时”模块
- D、加入“多电机驱动”模块和“延时”模块

5. 什么是可视化程序设计? 说一说你的理解。

6. 编写 VPL 程序, 指挥“学习者号”完成“官方作品”文件夹中的“新手入门\_前进”仿真任务。

7. 编写 VPL 程序, 指挥“学习者号”完成“官方作品”文件夹中的“新手入门\_后退”仿真任务。

8. 编写 VPL 程序, 指挥“学习者号”完成“官方作品”文件夹中的“新手入门\_向左走”仿真任务。

9. 编写 VPL 程序, 指挥“学习者号”完成“官方作品”文件夹中的“新手入门\_向右走”仿真任务。



## 综合实践活动

1. 在 VPL 程序编辑窗口编写程序“vpl\_一鼓作气”, 指挥“学习者 3 号”完成“一鼓作气”仿真任务, 比一比谁的仿真分值最高。

2. 在机器人实验室, 6~8 个同学组成 1 个小组, 分别将“vpl\_一鼓作气”程序从电脑上下载到“学习者号”上, 在“一鼓作气”场地对“学习者号”中的传感器、程序进行调试、运行, 比一比哪个小组成绩最好(再次填写表 9.2, 统计各组成绩)。





## 学习评价

完成下列各题, 综合评估自己在知识与技能、解决实际问题的能力以及相关情感态度与价值观的形成等方面, 是否达到了本学期的目标。

1. 用“画图”程序绘图和用 LOGO 程序绘图的主要区别是什么? 各有什么优势?
2. 我学过的 LOGO 命令有: \_\_\_\_\_;  
最擅长使用的 LOGO 命令是: \_\_\_\_\_。
3. LOGO 除了有画画功能外, 还有\_\_\_\_\_功能。
4. 在\_\_\_\_\_情况下, 可以使用 REPEAT 命令提高画图效率?
5. 建立过程的原因是: \_\_\_\_\_。
6. 过程由 \_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_3 部分组成。带参数过程和不带参数过程的适用范围分别是: \_\_\_\_\_。
7. 程序设计的一般过程是: \_\_\_\_\_。
8. 机器人的三大核心部件是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
9. 在仿真平台中编写 LOGO 程序, 并结合真实环境中机器人运行的实际情况进行调试, 编程的一般方法步骤为: \_\_\_\_\_。
10. 因为机器人是由一个个零部件组装成的, 所以在搭建机器人的过程中, 根据任务需求, 总是先组装\_\_\_\_\_, 再组装\_\_\_\_\_。
11. 在学习过程中, 我使用了\_\_\_\_、\_\_\_\_、\_\_\_\_和\_\_\_\_传感器, 它们的功能和特点的分别是\_\_\_\_、\_\_\_\_、\_\_\_\_和\_\_\_\_\_。
12. 算法是\_\_\_\_\_, 我们学了两种表示算法的方法, 它们是: \_\_\_\_\_和\_\_\_\_\_。
13. 在 LOGO 中, \_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_语句可以实现程序的循环结构, 它们的适用范围是: \_\_\_\_\_。
14. 在 IRobotQ3D 中, 有\_\_\_\_\_和\_\_\_\_\_两种程序设计语言可以指挥机器人完成各种各样的任务。
15. 用程序指挥机器人完成一个复杂的大任务时, 总是将其分解成若干小任务, 对每个小任务编写程序, 从而构成一个大任务的程序, 这种思想被称作\_\_\_\_\_。
16. 我觉得, 要想学好 LOGO 与机器人, 不仅要掌握信息技术的相关知识, 还要掌握\_\_\_\_\_等方面的相关知识。
17. 通过对 LOGO 与机器人的学习, 我最大的收获是: \_\_\_\_\_。
18. 在学习 LOGO 与机器人的相关知识的过程中, 我难以理解的内容有: \_\_\_\_\_。
19. 除了书本知识, 我还学会了: \_\_\_\_\_, 我还想知道: \_\_\_\_\_。